

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Ould-Rouis Meriem Ouslimani Djedjigua

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

Sujet :

***Une architecture orientée service et d'aide à la décision dédiée au E-Santé.
Application : Le diabète.***

Promotrice : Mme. Boumahdi.F

Promotion 2015 _ 2016

Résumé

L'objectif appréhendé par le présent travail, est automatiser le processus de traitement d'un malade infecté par le diabète insipide à l'aide d'une architecture orientée services (SOA) afin de faciliter la tâche et la communication entre un médecin spécialiste en Endocrinologie, testeur et un pharmacien ; afin de nous aide même à soigner les malades éloignés géographiquement. Dans ce cadre, nous avons utilisé le diagramme de cas d'utilisation, diagramme DMN et également le diagramme BPMN, ce dernier a été utilisé pour l'exécution et la construction de nos services qui s'exécutés selon deux décisions : la première décision c'est la précision du cas du patient, s'il est infecté par le diabète insipide ou non. La deuxième décision c'est la détection du type du diabète insipide sachant qu'il existe trois types, cette décision permette au malade de prendre le traitement et au pharmacien de recevoir l'ordonnance pour terminer le long du processus.

Mot clés : Architecture orienté service, BPMN, DMN, services décisionnels, SOAML, le contrat de service.

Abstract

The goal apprehended by the present work is to automate the treatment process of a patient infected by diabetes insipidus with a service oriented architecture (SOA) to facilitate the task and communication between a specialist in Endocrinology, tester and a pharmacist; so we even help treat patients geographically distant. In this context, we used the use case diagram, chart and also DMN BPMN diagram, it has been used for the implementation and construction of our services which run in two decisions: the first decision c is the patient's case of accuracy, if infected with insipid diabetes or not. The second decision is the detection of diabetes insipidus like knowing that there are three types, this decision allows the patient to take the treatment and the pharmacist receiving the order to complete the entire process.

Key words: service oriented architecture, BPMN, DMN, decision services, SoaML, the service contract.

REMERCIEMENTS

*Nous tenons, avant tout, à remercier ALLAH, tout clément, tout puissant,
de nous avoir donné la force de réaliser ce travail.*

*Notre sincère reconnaissance et notre sincère gratitude à Madame
BOUMAHDI Fatima, notre promotrice, pour nous avoir proposé le sujet, pour
sa disponibilité, pour ses précieux conseils, sa confiance qu'elle nous a toujours
témoigné et sa sollicitude dont elle nous a entouré, et ce tout au long de
l'élaboration du présent travail.*

*Nous exprimons notre profonde reconnaissance aux membres du jury pour
avoir accepté de juger notre travail.*

*Que tous nos professeurs qui ont contribué à notre formation trouvent ici
notre plus profonde gratitude.*

Dédicaces

*Je dédie du plus profond de mon cœur ce
modeste travail,*

*À ma très chère mère « NACIMA », pour toi je dédie toute ma
vie,*

*À mon adorable père « HAMID », merci pour ton soutien à
Toute épreuve,*

À mes chères sœurs « KARIMA » « SARA » et « YASMINE »

À mes chers grands parents,

À tout les membres de ma grande famille,

À ma binôme « DJEDJIGUA » et sa famille,

À tous mes amis,

À tous les collègues de promotion,

À tous ceux qui m'ont soutenu.

Ould-Rouis Meriem

Je dédie ce travail

A la mémoire de ma grande mère « ouezna »

*A mes très chers parents qui ont sacrifié leur vie pour
ma réussite, et pour leurs encouragements. Que dieu*

les garde éternellement heureux.

*A mes chers sœurs : ouezna, sonia et son marie hakim et le petit
ange imade edin .*

*A ma chère amie, mon binôme Meriem qui m'a supporté durant
cette année .et qui j'ai trouvé l'entente dont j'avais besoin.*

A tous mes amis.

A ma meilleur amie Hala et sa famille.

A mes grandes sœurs : ghania, nesrine, aicha.

Ouslimani Djedjigua

Sommaire

I.Introduction générale	2
1. Introduction	3
2. Présentation du sujet	3
3. Organisation du mémoire	4
II .Chapitre I : l'architecture orienté service	6
1. Introduction.....	7
2. Architecture d'un système d'information	8
3. L'orientation service.....	10
4. La composition et l'orchestration des services.....	12
5. Les éléments de base de l'architecture.....	13
6. Les méthodes de développement d'une architecture SOA.....	16
7. Comparaison entre les méthodes de mise en place d'une SOA.....	21
8. Avantages et inconvénients d'une architecture orientée service.....	23
9. L'architecture d'entreprise.....	24
10. L'E-santé.....	26
11. Conclusion	28
III.Chapitre II : les langages de modélisation	29
1. Introduction.....	30
2. Les langages de modélisation.....	31
3. Description globale de la méthode proposée SOA+d	39
4. Description détaillée de la méthode proposée SOA+d	40
5. Conclusion.....	46
IV.Chapitre III : La conception	47
1. Introduction	48
2. Phase 1 : Analyse.....	48
3. Phase 2 : Identifier et classer les services.....	58
4. Phase 3 : Modélisation des services.....	69
5. Phase 4: Implémentation du système	73
6. Conclusion.....	79
V.Conclusion générale	80
1. Conclusion.....	81
VI.Annexe	84
VII.Bibliographie	104

Table des figures

Figure I-1 Architecture d'un d'information.....	9
Figure I-2 L'autonomie et l'interopérabilité (Fournier-Morel,Grosjeans, Plouin, Rognon,2011).....	12
Figure I-3 L'orchestration et la chorégraphie (boumahdi, 2015).....	13
Figure I-4 Architecture d'un bus ESB (David Chappell, 2015).....	15
Figure II-1 L'utilisation des standards par SOA+d (boumahdi, 2015).....	30
Figure II-2 Diagramme de participants (boumahdi, 2015).....	33
Figure II-3 Un diagramme d'interface de service (boumahdi, 2015).....	34
Figure II-4Exemple de Contrat de service (boumahdi, 2015)	34
Figure II-5 Exemple d'architecture de services (boumahdi, 2015)	35
Figure II-6 Exemple de données de services (boumahdi, 2015).....	36
Figure II-7Capacités de services nécessaires pour le traitement des commandes d'achat (boumahdi, 2015).....	36
Figure II-8Les phases des SOA+d (boumahdi, 2015).....	40
Figure II-9 La description des phases de SOA+d avec la notation BPMN (boumahdi, 2015)	41
Figure II-10 Les choix technique pour implémenter une architecture SOA (boumahdi, 2015).....	44
Figure III-1Diagramme de cas d'utilisation.....	48
Figure III-2 Le diagramme BPMN de notre processus E-santé.....	51
Figure III-3Classement des patients DMN.....	53
Figure III-4La représentation logique de la prise de décision 'Classement des patients'.....	55
Figure III-5 Le classement des malades selon leurs types du diabète insipide DMN.....	56
Figure III-6 La représentation logique de la prise de décision 'Classement Diabétique'.....	57
Figure III-7Les services de la vision Métier.....	66

Figure III-8 Les services de la vision «Système Information».....	67
Figure III-9 Les services de la vision «Décision».....	68
Figure III-10 Architecture générale.....	69
Figure III-11 Le contrat de service Réception des patients.....	70
Figure III-12 Le contrat de service Diagnostic des patients.....	70
Figure III-13 Le contrat de service Classement des patients.....	71
Figure III-14 Le contrat de service Résultat des tests.....	71
Figure III-15 Le contrat de service Classement Diabétique.....	72
Figure III-16 Le contrat de service Traitement des malades.....	72
Figure III-17 Choix techniques.....	73
Figure III-18 Une vue globale de la plateforme de destination.....	74
Figure III-19 Vue arborescente des services d’infirmier.....	75
Figure III-20 Vue arborescente des services du médecin.....	75
Figure III-21 Vue arborescente des services d’examineur.....	75
Figure III-22 Vue arborescente des services du pharmacien.....	76
Figure III-23 Diagramme de séquence entre l’infirmier et le médecin.....	77
Figure III-24 Diagramme de séquence entre le médecin et l’examineur.....	77
Figure III-25 Diagramme de séquence entre l’examineur et le médecin.....	78
Figure III-26 Diagramme de séquence entre le médecin et le pharmacien.....	79

Liste des tableaux

Tableau I-2 Les avantages et les inconvénients du SOA.....	24
Tableau II-1 Les deux syntaxes concrètes graphiques proposées par SoaML.....	32
Tableau III-1 Règle de passage 1.....	58
Tableau III-2 Règle de passage 2.....	59
Tableau III-3 Règle de passage 3.....	59
Tableau III-4 Règle de passage 3.....	60
Tableau III-5 Règle de passage 4.....	60
Tableau III-6 Règle de passage 4.....	61
Tableau III-7 Règle de passage 4.....	61
Tableau III-8 Règle de passage 4.....	62
Tableau III-9 Règle de passage 5.....	62
Tableau III-10 Règle de passage 5.....	63
Tableau III-11 Règle de passage 6.....	63
Tableau III-12 Règle de passage 6.....	64
Tableau III-13 Règle de passage 7.....	64
Tableau III-14 Règle de passage 7.....	65
Tableau III-15 Règle de passage 9.....	65
Tableau III-16 Règle de passage 9.....	66

Introduction Générale

1. Introduction

Une architecture orienté service est une architecture logicielle s'appuyant sur un ensemble de services simples, qui sont développés en s'inspirant des processus métier de l'entreprise. Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services.

La SOA (Service Oriented Architecture) doit dépasser le cadre technique lié au niveau l'informatique et au niveau métier pour toucher le niveau décisionnel de l'entreprise. Dans ce sens, le vrai challenge consiste à étendre la SOA à l'aspect décisionnel de l'entreprise. Le but de ce sujet, est d'ouvrir la boîte noire de la vision décisionnelle, et montrer ses composants selon une architecture SOA pour le processus « E_Santé ».

2. Présentation du sujet

2.1. Problématique

Dans ce sujet, nous avons choisis le processus « E_Santé », ce dernier inclut des problèmes de décisions.

La motivation principale de ce sujet est d'offrir l'aide pour résoudre les problèmes de décision dans le processus de E-Santé, en se basant sur une architecture orientée service.

Le processus d'E-Santé rencontre quelque problème au sein du système d'information comme :

- Le classement des patients selon leurs cas si sont infectées par le diabète insipide ou non.
- Le classement des malades selon leurs types du diabète insipide.
- L'obtention des résultats automatiquement, après la prise de décision.
- La décision prend beaucoup de temps et surtout la détection du moteur d'orchestration idéal qui a pris plus beaucoup du temps, et surtout pour la détection de la version approprié de Bonita.

2.2. Objectifs de l'étude

Le processus d'E-Santé peut engager dans le monde de SOA avec les objectifs suivants :

- Automatiser le classement des patients par rapport à leurs cas si sont infectées par le diabète insipide ou non.
- Aider le médecin à prendre des bonnes décisions concernant le type du diabète insipide.
- Connaitre à tout moment tous les informations concernant le patient.
- Une connexion entre le médecin, l'examineur et le pharmacien.
- Informatiser la communication entre le médecin et l'examineur et le pharmacien.
- Aider les malades à distance, et surtout ce qui souffrent des maladies chroniques.
- La réalisation d'un ensemble de services dans le domaine de la santé.
- La réalisation d'une application demandée par un médecin spécialiste en ENDOCRINOLOGIE.

3. Organisation du mémoire

Ce mémoire est composé d'une introduction générale suivie de deux parties, une partie théorique et une autre pratique, tout cela a été réalisé afin de résoudre et de faciliter la tâche pour un médecin spécialiste en ENDOCRINOLOGIE.

La première partie sera consacrée à l'étude théorique des grands concepts rencontrés lors de la réalisation de notre projet, structurée en deux chapitres.

Le premier chapitre « Introduction au architectures orientées services » dans lequel nous introduisons la notion de processus métier. Le second chapitre présente les méthodes utilisées dans la conception dans le domaine de la médecine.

La deuxième partie « La conception de l'architecture orientée service » consacrés d'abord à la conception de la solution pour notre problématique avec la méthode Soa+d, tout en définissant les différents aspects de cette méthode, en utilisant les langages de modélisation SOAML (Service Oriented Architecture Modeling Language), BPMN (Business Process Modeling Notation) et DMN (Décision Model and Notation). Ensuite, nous passerons à l'implémentation de la solution retenue au niveau de la conception après une description de l'environnement du travail. Nous achèverons cette partie avec la présentation du système réalisé, avec une vue générale sur l'application qui a pour but de résoudre un problème dans le domaine de E_santé de manière générale, et pour le diabète insipide de manière plus précise.

Sans oublier le rôle jouer par notre moteur Bonita dans le domaine d' E_santé et les avantages de cette réalisation.

Avec une conclusion générale résumera les principales leçons tirées de ce travail et les extensions possibles dans le futur.

À la fin, nous avons lié notre mémoire avec un article scientifique (annexe) qui résume tous notre travail avec la présentation de notre application, Le papier sera soumis à une conférence internationale (ISIA 2016).

Chapitre I:

L'architecture orientée services

1. Introduction

L'architecture orientée services est plus communément connue sous le sigle SOA pour *Services Oriented Architecture*. Cette notion est de plus en plus en vogue. Et tout laisse à penser que ce n'est pas prêt de s'arrêter.

Lorsque nous posons cette question, la réponse la plus courante est : "*les services web*". Or, je trouve que cette réponse est très réductrice. C'est comme si nous définissions une architecture client/serveur par le TCP/IP. En effet, les services web ne sont qu'un moyen de mettre en place une architecture orientée services et d'atteindre ses objectifs. Mais la SOA ne se limite pas qu'à une simple technologie.

Il est très difficile de définir la SOA en quelques mots. Ce que nous ne pouvons retenir, c'est que la SOA permet de répondre au problème du fonctionnement en silo d'une entreprise (suite aux multiples fusions, rachats, sites différents, évolutions technologique, etc...) qui empêche de gérer le SI d'une société dans sa globalité.

La SOA est une autre façon d'intégrer les briques ou composants logiciels au sein du SI de l'entreprise. Cette méthode s'attache à décomposer chaque composant en un ensemble de fonctions basiques appelées **services**. L'objectif étant de pouvoir réutiliser simplement ses services. Les utilisateurs de ceux-ci peuvent être d'autres services, des logiciels de l'entreprise, des applications externes, etc...

Dans ce chapitre, nous commençons par une définition de l'architecture orientée service, ensuite nous présentons les niveaux d'abstraction de SOA et ces méthodes de développement avec une comparaison entre les méthodes et aussi nous présentons les avantages et les inconvénients de SOA

Dans un deuxième temps, nous parlons sur l'architecture d'entreprise et la relation entre l'entreprise et le SOA.

En fin, nous présentons l'E_santé .

2. Architecture d'un système d'information

L'architecture informatique est à la base de tout système d'information : elle va le structurer et l'organiser par rapport aux fonctionnalités, aux attributs et aux composantes qui sont reliés entre eux. C'est dans cette logique, que les composantes sont modélisées, les relations sont décrites et les attributs sont documentés. Ainsi, l'architecture informatique va définir les règles nécessaires à la construction et à l'implémentation du SI.

Dans (Morley et al, 2005) (Boumahdi, 2015), une architecture du système d'information a été identifié comme « une représentation abstraite des différentes parties du système d'information qui permet des décisions globales et de s'assurer de la pertinence de l'assemblage, notamment la cohérence et l'efficacité technique. . . ».

Il existe une typologie riche d'architectures de système d'information. Ces architectures présentent des niveaux d'abstraction différents d'un système d'information. Nous distinguons : l'architecture fonctionnelle, logique et l'architecture physique.

2.1 L'architecture fonctionnelle

Elle représente des fonctions issues de l'analyse des processus métier. Exemple :

Gestion des Comptes. Elle permet de spécifier les besoins fonctionnels des utilisateurs du système d'information. Ainsi, un système d'information est présenté en termes de fonctions, liées entre elles par des flux d'information et liées aux acteurs qui les manipulent. Cette architecture intervient directement pour faire évoluer ou refondre un système d'information (Boumahdi, 2015).

2.2 L'architecture informatique (ou physique)

Elle décrit la structuration d'un système informatique en termes d'organisation de fonctions et des constituants qui le composent. Cette architecture définit d'autres « sous-architectures » :

L'architecture logicielle : permet l'agencement et l'interaction des composants logiciels. Ces derniers peuvent être définis sur plusieurs couches (couche présentation, couche applicative, couche données).

L'architecture matérielle : permet l'agencement et l'interaction des composants physiques (disque dur, unité centrale, etc.).

L'architecture d'intergiciel (Middleware) : permet l'agencement et l'interaction des composants servant à faire communiquer plusieurs applications entre elles.

L'architecture réseau : permet la communication au sein d'un système d'information (Boumahdi, 2015).

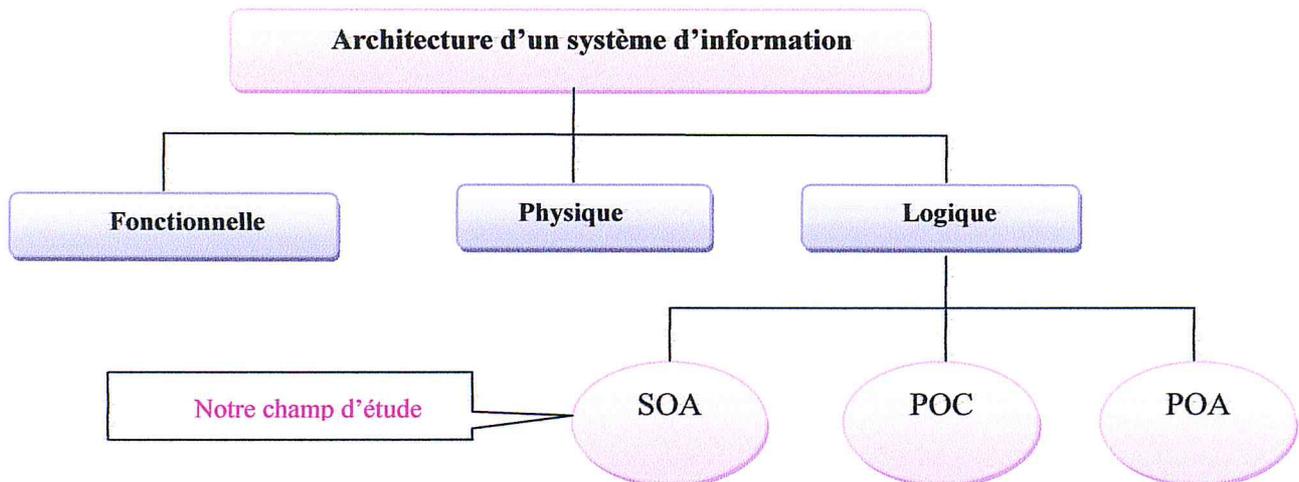


Figure I-1 Architecture d'un système d'information.

2.3 L'architecture logique

C'est la représentation virtuelle d'une architecture, abordable aux interlocuteurs. Nous pouvons aussi parfois parler d'architecture applicative logique.

Elle permet d'identifier la structuration d'un système d'information en adoptant une logique indépendante des considérations techniques. Les architectures logiques les plus connues sont : l'architecture orientée composants et l'architecture orientée services (SOA). Les composants ou les services sont caractérisés par le fait qu'ils sont des entités autonomes, réagissent par échange de messages et sont définis sur différents niveaux de granularité. La différence principale entre un service et un composant réside dans le fait qu'une architecture orientée services se concentre sur les exigences déterminées au niveau de la stratégie et du processus métier (nous parlons de service métier), alors qu'une architecture orientée composants se concentre sur les composants d'un logiciel (nous parlons de composant logiciel). Ces composants peuvent être utilisés pour livrer des services. Un service expose dans ce cas un composant (Boumahdi, 2015).

Le choix d'un style d'architecture est une décision très importante dans la conception de tout système informatique. Le but est de choisir la solution la plus simple et la plus économique en ressources, en temps de réalisation et aussi en temps d'exécution. Ce choix a

des conséquences directes sur la performance du système, sur la réutilisation et sur l'interconnexion avec d'autres systèmes.

Les architectures du système d'information ont un rôle capital pour la définition d'un système d'information et pour maintenir son évolution. Il existe donc naturellement un lien fort entre l'architecture du système d'information (conception du SI) et la capacité de prendre des meilleures décisions que ce dernier peut avoir.

3. L'orientation service

Défini comme étant la composante fondamentale de cette approche, le service permet de relier les consommateurs et les fournisseurs sur une même interface, partie visible de l'architecture par les utilisateurs.

Afin d'atteindre des objectifs stratégiques, l'entreprise est amenée à flexibiliser ses systèmes d'informations et ainsi développer des services. Afin de maximiser la flexibilité du système, les services doivent respecter quatre principes essentiels de construction tels que l'autonomie, la réutilisabilité, l'interopérabilité et la composition des services.

3.1 La réutilisation des services

Le principal but de l'architecture orientée service est explicité par ce principe de réutilisation de services dans les processus du système.

Le but étant de créer des architectures favorisant des modèles de systèmes ayant des composantes réutilisables. En effet, dans toute création de système, nous tentons de créer des fonctionnalités répondant à des critères précis. Chaque fonctionnalité est spécifique à une tâche précise. Dans cette logique, l'utilisation d'une fonctionnalité d'un service pour une autre tâche est quasi impossible si nous tenons compte également du manque de communication entre les services. (Erl, 2008, pp. 254-259).

L'implémentation de la SOA offre au service, la possibilité d'être utilisé plusieurs fois, pour des tâches différentes et des utilisateurs différents.

Ainsi, les processus s'exécutent de manière continue dans le système et permettent d'atteindre les objectifs visés par la firme plus rapidement (Erl, 2006, p. 292).

3.2 L'autonomie et l'interopérabilité des services

Afin de favoriser la réutilisation des services, il est primordial que chacun de ces services soit autonome et capable d'interagir avec les autres composantes de l'architecture du système.

En général, les services sont très restreints au niveau de leurs champs d'exécution. En dotant un service d'une certaine autonomie, ce dernier devient capable de gérer ses ressources sans intervention externe (Erl, 2006, pp. 303-304).

L'autonomie leur confère un contrôle significatif sur leurs champs d'application.

Les services métiers doivent être indépendants les uns des autres, donc faiblement couplés. (Bonnet, Detavernier, Vauquier, 2008, p. 54).

Toutefois, les services ne sont que très faiblement reliés entre eux, empêchant ainsi une interaction au sein du système. Ceci engendre des problèmes de communication entre les diverses composantes et empêche le fonctionnement des processus en continue à cause des redondances, des duplications, des erreurs, etc.

Pour pallier à ces problèmes, il est primordial d'établir une communication entre les services, qui doivent interagir entre eux. Pour établir une communication cohérente, cette architecture implémente les services échangeant entre eux par messages et dans un langage standardisé (très souvent XML). L'utilisation d'un vocabulaire métier uniformisé pour chaque service permet l'intégrité du système. (Bonnet, Detavernier, Vauquier, 2008, p. 83).

Les échanges entre les différentes composantes du système, favorisent l'autonomie de ces composantes et permettent plus de flexibilité au système. (Erl, 2008, pp. 294-298).

3.3 La composition des services

Afin d'augmenter la réutilisabilité des services, il faut les créer de manière composée.

Chaque service, composé d'unité modulaire, doit être regroupé par niveau d'abstraction afin de séparer des logiques différentes telles que processus, métier ou présentation.

Construite de manière composée, donc par l'assemblage de multitudes d'unités modulaires et autonomes, les processus deviennent accessibles et compréhensibles par tous.

Une commande passée par un service est interconnectée à tous les autres services dont la présence est nécessaire pour accomplir la tâche en question.

De ce fait, un service qui fait partie intégrante d'une composition, se situe dans un contexte de partage et d'échange avec les autres services. Il doit être pris en compte avec toutes les autres composantes de l'application (Erl, 2008, pp. 388-392).

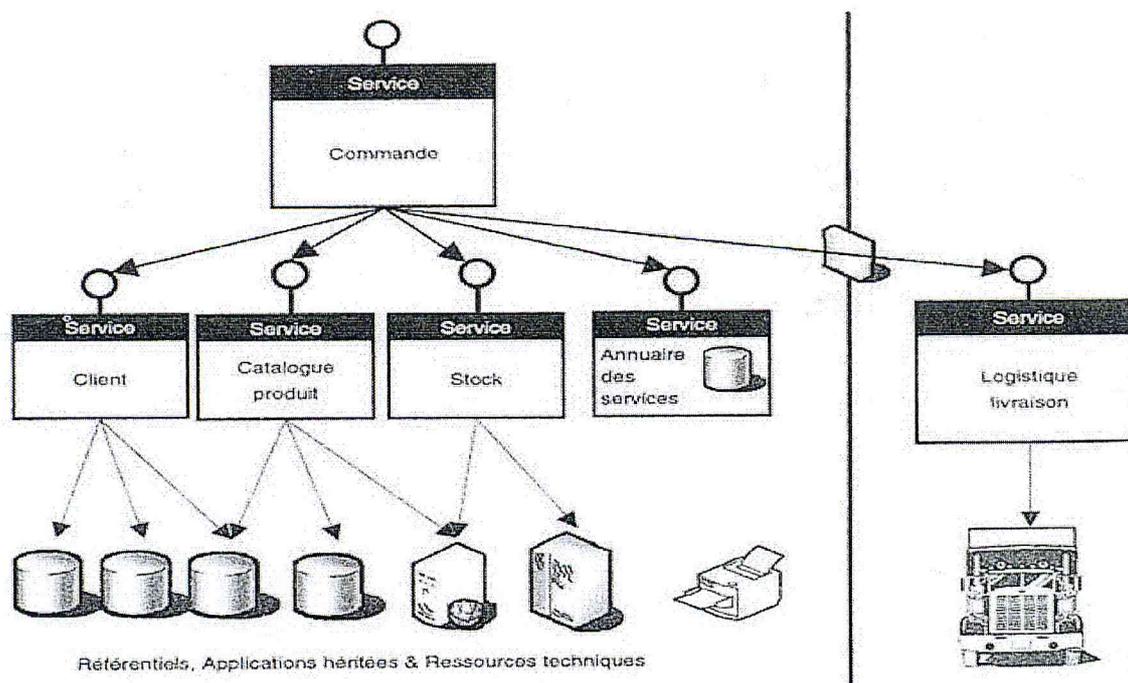


Figure I-2 et interopérabilité (Fournier-Morel, Grosjeans, Plouin, Rognon, 2011).

4. La composition et l'orchestration des services

La composition des services englobe les fonctionnalités nécessaires pour l'agrégation de différents services en un seul service composite pouvant être utilisé soit directement par les clients soit dans la composition de nouveaux services. Une fois composée, l'exécution de la chaîne de services peut être faite grâce à :

L'orchestration : qui permet de spécifier l'ordre d'exécution des services et de décrire le flot de contrôle du processus qui sera géré par un moteur d'exécution (Boumahdi, 2015).

L'orchestration suppose un contrôleur principal pour la composition des services et que les messages envoyés entre les activités ne sont pas visibles de l'extérieur.

En d'autres termes, l'orchestration repose sur l'interaction entre les services qui composent un processus.

La chorégraphie : se place dans une vision plus collaborative dans laquelle plusieurs processus échangent des messages en ayant une vue commune des différentes interactions, des différents messages et des activités de chaque partenaire (Boumahdi, 2015).

Cette vue commune a été préalablement définie par les partenaires, selon ce que chaque partenaire a déclaré être capable de faire. C'est donc une vue externe ou publique.

La **Figure I-3** illustre la distinction entre la chorégraphie et l'orchestration. Dans ce cas-ci, il y a deux organisations ayant chacune un processus métier distinct. Par exemple l'orchestration pour la commande d'une pièce et l'orchestration pour le traitement d'une commande de pièce. La chorégraphie serait l'intégration des deux processus. Si on suit cet exemple, l'action de commander une pièce déclenche automatiquement le processus de traitement d'une commande.

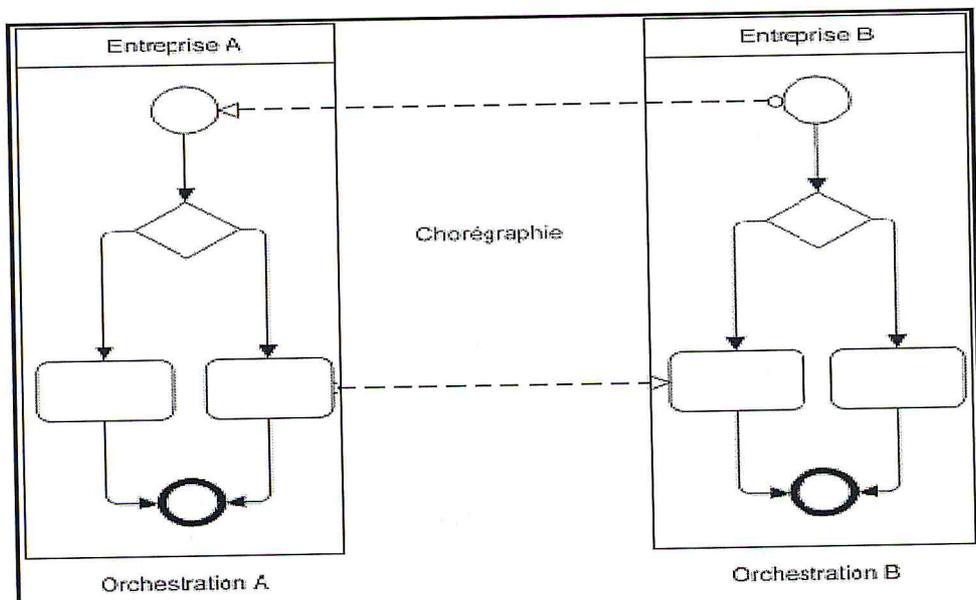


Figure I-3 L'orchestration et la chorégraphie (Boumahdi, 2015).

5. Les éléments de base de l'architecture

Nous pouvons résumer les éléments de base de l'architecture comme suit:

5.1 Composant de service

Le composant de service est la brique de base de l'architecture. Il se décompose en deux parties : la vue externe (ou spécification de service), qui expose la facette service proprement dite, et la vue interne, qui décrit le contenu du composant. La vue externe est constituée par

un ensemble d'opérations de service regroupées en interfaces (au sens UML), et de l'appareillage pour les utiliser (types des données échangées, contrat de service, propriétés, etc....). Cette spécification est décrite en général par un fichier WSDL ou équivalent.

La vue interne contient des informations relatives à la logique interne comme le détail des traitements ou les bases de données manipulées. Nous trouvons également les références vers les services utilisés par le composant. Cette vue est masquée aux consommateurs du composant de service. Elle est employée notamment par les architectes SI, qui travaillent sur la vision globale du système.

Les composants de service sont identifiés et définis avant tout suivant une logique métier, indépendamment de la technologie. Nous distinguons pour ce faire le composant logique, qui fixe la structure du système dans le cadre de l'architecture logique, et le composant logiciel, qui est sa traduction technique déployée (Raymond, 2011).

5.2 Bus d'entreprise

Le système est constitué d'un ensemble de participants communiquant qui jouent le rôle de consommateur et de fournisseur de service. Les consommateurs de service peuvent être divers : des applications, progiciels ou d'autres composants de service.

Les composants de service sont les fournisseurs de service du système, dans lesquels on distingue deux familles: les composants qui prennent en charge directement l'implémentation des services et ceux qui délèguent cette implémentation à un tiers (mainframe, ERP, application existante). Il faut néanmoins préciser que dans ce cas de figure, le composant n'est pas toujours assimilé à un simple passe-plat. Des traitements d'adaptation sont fréquemment nécessaires (format des données, encapsulation de service) afin de mieux intégrer les besoins des consommateurs de service.

Le bus d'entreprise (ESB) agit comme la colonne vertébrale reliant ces participants d'une manière banalisée à travers les interfaces de services. Cela permet notamment les modifications d'implémentation ou le remplacement des « legacy » sans remanier la structure de fonctionnement du système (Raymond, 2011).

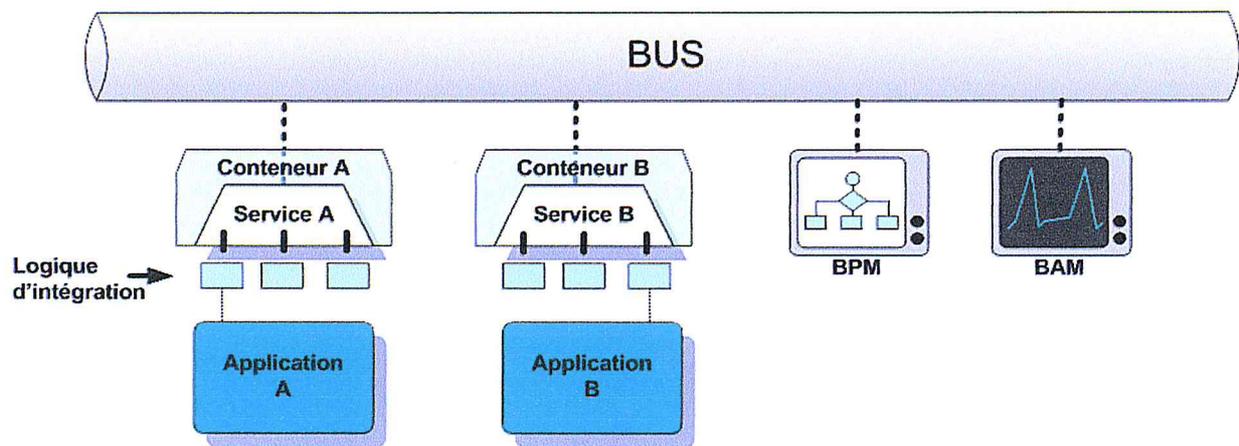


Figure I-4 Architecture d'un bus ESB (David Chappell, 2015).

5.3 Contrat de service

Le contrat de service joue un rôle majeur : il détaille les conditions d'utilisation du service sous forme de pré et post conditions, protocoles, et contraintes (QoS,SLA). Les contraintes non fonctionnelles permettent de fixer les termes du contrat opérationnel entre consommateur et fournisseur de service.

Le protocole d'utilisation d'un composant de service définit les séquences valides d'invocation des ses opérations. A noter que les composants de service ne sont pas des objets : ils ne conservent aucun état (une seule instance du composant est disponible). Les conditions portent sur des attributs des données d'échange transmises lors de l'invocation des opérations (Raymond, 2011).

5.4 Données d'échanges et données persistantes

La distinction entre les données d'échange et les données persistantes est inhérente aux architectures SOA, qui isolent les bases de données à l'aide de services d'accès. Les données d'échange sont les informations véhiculées entre les participants (consommateurs ou fournisseurs de service) à travers l'invocation des opérations de service. Les données persistantes sont les informations contenues et gérées dans les bases de données. Ces informations sont structurées de façon habituelle (par exemple SGBD en mode relationnel), dans le cadre de référentiels ou de bases applicatives(Raymond, 2011).

5.5 L'annuaire de services

L'annuaire de services référence l'ensemble des services (et des contrats associés) disponibles au sein du SI, il participe ainsi activement à la mise en œuvre d'une cartographie

dynamique du SI. Dans un modèle de bus, l'annuaire peut être auto-alimenté par le service (enregistrement). Les annuaires UDDI forment aujourd'hui le standard de référencement des services (Raymond, 2011).

6. Les méthodes de développement d'une architecture SOA

Aujourd'hui, le monde industriel et académique s'intéresse de plus en plus à la modélisation des systèmes orientés services indépendamment des plates-formes technologiques ou des langages de programmation. La modélisation est d'une importance capitale pour le développement des SOAs (Papazoglou et al, 2007) et (Boumahdi, 2015) dans la mesure où elle définit les éléments de modélisation capables de représenter de tels systèmes à un haut niveau d'abstraction.

Plusieurs méthodes de mise en place d'une SOA ont été proposées dans la littérature.

Nous allons exposer dans ce qui suit une revue de la littérature des travaux que nous avons aidés pour proposer notre méthode.

6.1 SOMA: Service-Oriented Modeling and Architecture

La méthode d'IBM SOMA est proposée par (Arsanjani et al, 2008), elle couvre l'identification, la modélisation et la spécification des services.

La méthode SOMA comporte trois phases :

1. La phase d'identification des services se base sur trois démarches :

- une démarche descendante (Top-Down) pour la décomposition des domaines.
- une démarche ascendante (Bottom-Up) pour l'analyse du patrimoine existant.
- une démarche hybride (Middle-out) pour une modélisation du service selon le but.

2. La phase de spécification commence par filtrer l'ensemble des services candidats en se basant sur des règles bien déterminées à savoir :

- l'alignement entre fonction et métier.
- l'élimination des redondances.
- la réutilisation des services par les différents processus métiers.

- la facilité de l'implémentation.

Dans cette phase, les auteurs proposent d'attribuer une note pour les services sur une échelle de 1 à 5.

Les services qui auront les scores les plus élevés seront considérés comme des candidats pour la réalisation.

3. La phase de réalisation se charge d'implémenter l'ensemble des services.

L'approche SOMA (Arsanjani et al, 2008) et (Boumahdi, 2015) est une approche à la fois descendante et ascendante qui tient compte des besoins métier et de l'existant de l'entreprise.

Cependant SOMA ne propose pas une description ouverte de sa méthode ce qui rend difficile l'analyse de ses capacités réelles.

(Papazoglou et al, 2007) Ont proposé une méthode de développement des services nommée

eXtended SOA (xSOA).

6.2 xSOA: eXtended SOA

La méthode englobe six phases et se base sur le Rational Unified Process (RUP), le développement orienté composant et la modélisation des processus (BPM : Business Process Modelling). Cette méthode propose des directives pour la spécification des interfaces de services Web ainsi que des modèles de flux de services, de manière à maximiser la cohésion et minimiser le couplage. Les six phases identifiées par (Papazoglou et al, 2007) sont :

- La phase de planification.
- La phase d'analyse et de spécification.
- La phase de construction et de test.
- La phase de provisionning.
- La phase de déploiement.
- La phase d'exécution et de supervision.

Il s'agit par la suite d'évaluer le service prenant en compte ses performances d'exécution.

La méthode proposée par (Papazoglou et al, 2007) présente un cycle complet pour le développement des services. Néanmoins, cette méthode n'a pas été bien détaillée ce qui peut poser des questions sur sa faisabilité. En outre, les auteurs ne présentent pas une architecture de soutien à leur méthode et la relation entre service et processus métier n'est pas assez détaillée.

6.3 MSOAM : Mainstream SOA Method

Dans l'ouvrage de (Erl, 2010) et (Boumahdi, 2015) intitulé « Service-oriented Architecture : Concepts, Technology, And Design », (Erl, 2010) et (Boumahdi, 2015) présentent une méthode de définition des services qui couvre les deux premières étapes du cycle de vie de la construction d'une SOA à savoir : l'identification et la conception des services.

Le processus d'identification des services (applicatifs et métier) proposé par (Erl, 2010) est une approche descendante comportant douze phases à réaliser dans l'ordre. Dans ce qui suit, nous allons présenter ces différentes phases.

La première phase consiste à décomposer les processus métiers déjà cartographiés en un ensemble d'étapes. Une fois identifiées, il s'agit par la suite d'éliminer les étapes qui ne peuvent pas être encapsulées par des services telles que les étapes réalisées manuellement. Par la suite, la maîtrise d'ouvrage se charge d'identifier les services potentiels en regroupant les étapes qui semblent appartenir à un même contexte.

Chaque étape pourra correspondre à une opération au sein d'un service. Par la suite, (Erl, 2010) propose d'identifier les logiques métiers issues de l'étude des processus.

L'objectif d'une telle identification est de sélectionner les étapes éligibles au rang d'opérations de services.

La phase suivante du processus d'identification propose d'appliquer la logique de la SOA aux services potentiels préalablement définis.

L'auteur insiste tout particulièrement sur l'importance d'appliquer deux principes de base de la SOA aux services métier à savoir : la réutilisation et l'autonomie. L'auteur prévoit que l'analyste métier pourra ajouter des opérations à un service particulier afin d'améliorer sa réutilisation future.

La sixième phase du processus d'identification consiste à définir les services composés.

Ceci se réalise à travers la construction des différents processus métiers par composition des services. Ainsi, il sera plus facile de détecter le besoin de développer un nouveau service ou encore le besoin d'ajouter une opération à un service. Quant à la septième phase, elle se charge d'étudier les besoins fonctionnels de chaque opération d'un service.

Ces besoins seront pris en considération dans la huitième phase qui permet d'identifier les opérations des services applicatifs à partir des besoins d'implémentation exprimés dans la phase précédente. Par la suite, il s'agit de regrouper les opérations appartenant à un même contexte afin d'identifier les services applicatifs. Ces derniers vont être vérifiés dans la phase suivante afin de s'assurer qu'ils sont réutilisables et autonomes.

Les deux dernières phases du processus proposé par (Erl, 2010) consistent à revoir les services déjà identifiés ainsi que leurs opérations.

Le travail proposé dans (Erl, 2010) et (Boumahdi, 2015) traite l'identification des services d'entreprise.

L'approche proposée est une approche assez claire et comporte des exemples qui montrent chaque phase du processus d'identification. D'ailleurs ce point constitue l'un des points forts de l'approche. Cependant, le travail a été testé sur un exemple assez simple qui est un peu loin de la réalité des entreprises. De plus, la typologie de services n'est pas supportée par un méta-modèle qui récapitule l'ensemble des concepts manipulés ainsi que les relations entre eux. Ce qui laisse régner un certain flou concernant la signification de certains types de services (tels que par exemple les services hybrides). En outre, l'approche proposée est une approche descendante qui signifie une refonte de tout ou une partie du système d'information de l'entreprise, jugée bien souvent trop coûteuse et trop risquée.

6.4 SOD-M: Service-Oriented Development Method

propose une approche orientée service et le développement d'un modèle de base pour les systèmes web intitulée SOD-M, cette approche présente des modèles et des standards qui se focalisent principalement sur le développement des caractéristiques comportementales des systèmes, et l'établissement des standards pour la construction des modèles métier à un haut niveau d'abstraction (De Castro et al, 2011) (Boumahdi, 2015).

L'approche décrit trois méta-modèles organisés selon l'architecture MDA (Model Driven Architecture) (De Castro et al, 2011) (Boumahdi, 2015) et les niveaux d'architecture : CIM (Computational Independent Models), PIM (Platform Independent Models) et PSM (Platform Specific Models).

La méthodologie fournit un framework pour le développement des applications orientées services avec des modèles qui peuvent exprimer tout le processus de développement des applications basées sur les services. Cependant, SOD-M n'a aucun soutien pour la description et la modélisation des exigences non fonctionnelles.

6.5 La méthode Praxeme

Praxeme est une méthodologie d'entreprise qui propose des procédés de modélisation pour tous les aspects de l'entreprise pour décrire de façon pertinente et exhaustive la réalité sur laquelle on veut intervenir.

Elle propose un cadre de référence qui identifie huit aspects du Système Entreprise.

Ces aspects sont articulés rigoureusement dans la topologie du Système Entreprise.

Cette dernière fournit, au référentiel Praxeme, sa base théorique.

La Topologie du Système Entreprise proposée par Praxeme est un schéma qui articule les aspects suivant (Vauquier, 2007) (Vauquier, 2013) (Boumahdi, 2015):

- L'aspect sémantique.
- L'aspect pragmatique.
- L'aspect géographique.
- L'aspect Logique.
- L'aspect technique.
- L'aspect matériel du système.
- L'aspect logiciel.
- L'aspect Physique.

Actuellement, les développeurs de la méthode publique d'Architecture d'Entreprise

Praxeme pensent à l'intégration du nouveau standard DMN, qui sera utilisé pour formaliser les règles métier et organiser les indicateurs de performance (BIGAND, 2014) (Boumahdi, 2015).

6.6 SOA+d: Service Oriented Architecture with a Decision

la modélisation orientée services est la discipline des systèmes d'entreprise et de logiciels de modélisation, dans le but de concevoir et de spécifier les systèmes d'entreprise axée sur le service au sein d'une variété de styles architecturaux, tels que l'architecture d'entreprise, l'architecture de l'application, l'architecture orientée services, et le Cloud computing.

Toute méthodologie de modélisation orientée services comprend généralement un langage de modélisation qui peut être utilisé à la fois par l'organisation de domaine du problème (le commerce), et l'organisation de domaine de solution (le Département des technologies de l'information), dont les perspectives uniques influencer généralement la stratégie du cycle de vie de développement des services et les projets mis en œuvre en utilisant cette stratégie.

La modélisation orientée services vise généralement à créer des modèles qui fournissent une vue d'ensemble de l'analyse, la conception et l'architecture de tous les «Entités de logiciels» dans une organisation, qui peut être compris par des personnes ayant différents niveaux d'activité et de compréhension technique. La modélisation orientée services encourage généralement l'affichage des entités logicielles comme «actifs» (actifs axés sur les services), et fait référence à ces actifs collectivement comme des «services».

7. Comparaison entre les méthodes de mise en place d'une SOA

En se basant sur un ensemble de bonnes pratiques pour la mise en place d'une SOA telles que celles présentées dans (Mukhtar, 2013) et (Ramollari et al, 2007) et (Boumahdi, 2015) nous avons synthétisé les caractéristiques des différentes méthodes à l'aide de cinq critères :

Démarche de développement : Il existe trois démarches de développement d'une approche SOA. La première est l'approche descendante (D) dans laquelle la logique métier des processus existants est examinée afin d'identifier les services. À l'inverse, l'approche ascendante (A) débute par l'analyse du patrimoine applicatif afin de déterminer les fonctions existantes du système d'information et tracer ainsi une cartographie applicative. À partir de cette cartographie, il est possible d'identifier les fonctions qui sont éligibles au rang de

service. Enfin, l'approche hybride (H) préconise de mener en parallèle une approche descendante et une approche ascendante.

Documentation : les approches se différencient dans leur documentation. Ce critère est utilisé pour analyser quelles sont les approches qui ont déjà été appliquées dans la pratique et quelles sont celles qui ont été purement dérivées des bases théoriques.

Langages et notations de modélisation : ce critère permet de renseigner si la méthode propose de réutiliser des langages et les notations déjà existantes. Comme exemples de ces langages et les notations, nous pouvons noter la conception orientée objet (UML :), le développement basé sur le langage SoaML, les modèles de processus (BPMN : Business Process Modelling Notation) et le nouveau standard DMN (decision modeling and notation).

Prise en compte de la décision : la décision est une caractéristique importante et largement recherchée par les entreprises. Dans cette perspective, la décision se manifeste à travers la prise en compte de la solution (i.e., usage des services) au moment de leur modélisation. En outre, ce critère permet d'indiquer si les méthodes proposées considèrent les spécificités et les solutions des problèmes de décision. Nous avons choisi d'examiner les méthodes proposées par rapport à cette caractéristique.

Nous pouvons aussi présenter la comparaison entre les méthodes SOA par un tableau comme suit :

	Démarche	Les Standards utilisés	Documentation	Propriété	Décision
SOA Decision Modeling (SOAD) 2009 (Zimmermann, 2009)	H	Non	Exemple	Oui	Oui
Service-oriented modeling and architecture (SOMA) 2008 (Arsanjani, et al., 2008).	H	UML	Exemple	Oui	Non
A Platform Independent Model for SOA (PIM4SOA) (2006) (Benguria & Larrucea, 2006)	D	Non	Exemple	Non	Non
CBDI-SAE Meta Model (2011) (Everware-CBDI, 2011)	D	SoaML + UML profile	cas d'étude	Oui	Non
Modelling of Service Oriented Architectures With UML (2008) (Lopez-Sanz, et al., 2008)	D	UML profile	cas d'étude	Non	Non
A Service Oriented Information System: A Model Driven Approach (SOIS) (2012) (Fazziki, et al., 2012)	A	SoaML	Cas d'étude	Non	Non
SOD-M (the Service-Oriented Development Method) (2011) (De Castro, et al., 2011)	D	Non	Cas d'étude	Non	Non
Dynamic Homecare Service Provisioning Architecture (2012) (Zarghami, et al., 2012)	D	Non	Cas d'étude	Non	Oui
SOA rd : the Service-Oriented architecture with a decisional aspect (Boumahdi, et al., 2014)	D	SoaML+DMN+UML BPMN	Cas d'étude	Non	Oui

Tableau I-1 Comparaison des méthodes de mise en place de la SOA (Boumahdi, 2015).

8. Avantages et inconvénients d'une architecture orientée service

Nous allons présenter par la suite les avantages et les inconvénients de cette approche :

Avantages	Inconvénients
<ul style="list-style-type: none"> • Obligation d'avoir une modélisation poussée • Possibilité de découpler les accès aux traitements • Localisation et interfaçage transparents (ouverture accrue) • Possibilité de mise en place facilitée à partir d'une application objet existante • Réduction des coûts en phase de maintenance et d'évolution • Facilité d'amélioration des performances pour des applications importantes (répartition des traitements facilitée) 	<ul style="list-style-type: none"> • Coûts de conception et de développement initiaux plus conséquents • Nécessité d'appréhender de nouvelles technologies • Existant non SOA dans les entreprises • Performances réduites pour des traitements simples (couche supplémentaire)

Tableau I-2 Les avantages et les inconvénients du SOA (Gonzalez, 2015).

9. L'architecture d'entreprise

Pourquoi une entreprise a besoin d'une architecture SOA - ou une architecture de toute autre nature?

La fonction de direction d'une entreprise - le conseil d'administration d'une société commerciale, ou la gestion de haut niveau d'un département de division ou de gouvernement, par exemple - fixe des objectifs pour l'entreprise, et décide comment il devrait fonctionner pour les atteindre. Une architecture clairement articulée décrit l'organisation de l'entreprise souhaitée et le mode de fonctionnement. Ce faisant, il fournit:

- Une définition des changements qui devraient être mises en œuvre pour atteindre cet organisme.
- Une base pour le contrôle et la gouvernance de son fonctionnement continu.

Une architecture d'entreprise fournit également un troisième avantage. Les entreprises changent au fil du temps. Ils combinent et divisé, comme dans les fusions commerciales et spin-offs, ou les grands réorganisations gouvernementales. Il est plus facile de combiner une

entreprise avec une autre, ou de le scinder en composants, quand il a une architecture clairement définie. Cela apporte d'importantes économies de coûts, et peut augmenter la valeur d'une entreprise commerciale.

L'architecture d'entreprise dans son sens le plus large comprend beaucoup plus que IT. Il couvre les opérations commerciales, les finances, les gens et les bâtiments, en plus de la technologie, et il couvre d'autres technologies que l'informatique, par exemple pour la fabrication ou le transport. L'architecte d'entreprise doit comprendre ces zones, au moins assez bien pour superviser les architectes qui se spécialisent en eux. L'architecte informatique doit être capable de travailler en équipe avec de tels spécialistes.

La qualité de l'architecture informatique d'une entreprise peut avoir un impact majeur sur sa performance de l'entreprise. Depuis les années 1950, les organisations commerciales et gouvernementales sont devenues de plus en plus tributaires de l'informatique pour la conduite de leurs activités quotidiennes, et cette tendance semble devoir se poursuivre. Les entreprises qui utilisent prospèrent efficacement. Le meilleur des entreprises .com autrefois tourné en dérision «Quand seront-ils jamais faire un profit?» Sont devenus des noms familiers. Les entreprises ayant une mauvaise tomber derrière leurs concurrents, ou échouent.

En raison de son importance pour l'ensemble des activités, l'architecture informatique de l'entreprise est devenue une profession. Aucune entreprise ne songerait à entreprendre le développement d'un grand bâtiment sans engager un architecte de bâtiments avec un statut professionnel qui offre une garantie de compétence. De même, les entreprises qui entreprennent le développement de grands systèmes informatiques recherchent des architectes informatiques des entreprises professionnelles. Leur statut en tant que professionnels indique qu'ils comprennent, et ont un bilan de l'application, les meilleures méthodes et techniques de l'architecture informatique.

9.1 L'Architecture orientée services et l'entreprise

Un architecte d'entreprise se penche sur la construction globale de l'entreprise. SOA est une technique de construction particulière qui peut être utilisé pour construire l'entreprise IT.

Une technique particulière peut avoir un impact majeur sur la construction globale. L'introduction de techniques à ossature d'acier dans la dernière partie du 19ème siècle a révolutionné l'architecture des bâtiments. Elle a permis les gratte-ciel des années **1920**, et les bâtiments encore plus que nous avons aujourd'hui.

SOA pourrait avoir un impact similaire sur l'architecture informatique. Il ne pas augmenter la taille des systèmes informatiques, mais il n'augmente leur interopérabilité.

Avec SOA, les systèmes d'information fournissent des services qui sont définis et décrits dans le cadre des activités commerciales de l'entreprise. Chaque service est identifié, et ce qu'il fait est clairement défini sous la forme d'un contrat. Ce principe permet d'utiliser des techniques telles que la composition de services, la découverte, la communication à base de messages, et la mise en œuvre dirigée par les modèles, qui donnent le développement rapide de solutions efficaces et flexibles. Ils sont des caractéristiques importantes de la SOA. Leurs avantages - en particulier celle de l'agilité - sont les raisons les plus fréquemment cités pour l'adoption de SOA.

Mais il est le remplacement de grandes applications monolithiques qui ont des interfaces d'interopérabilité minuscules, à contrecœur fourni ne peut pas garantir, par les petits, les services modulaires qui ont des descriptions d'interface et des contrats, qui est l'effet le plus fondamental de la SOA. Ceci est la base de l'énorme augmentation de l'interopérabilité des systèmes d'information que la SOA peut apporter, non seulement au sein des entreprises, mais aussi entre les entreprises.

10. L'E-santé

Le terme d'e-santé (ehealth en anglais) désigne tous les aspects numériques touchant de près ou de loin la santé. Cela correspond à du contenu numérique lié à la santé, appelé également la santé électronique ou télésanté.

De manière plus générale, on trouve également ce terme pour expliquer l'application des technologies de l'information et de la communication à l'ensemble des activités en rapport avec la santé dans son acceptation la plus large.

Cela concerne des domaines comme la télémédecine, la prévention, le maintien à domicile, le suivi d'une maladie chronique à distance (diabète, hypertension, insuffisance cardiaque ...), les dossiers médicaux électroniques ainsi que les applications et la domotique, en passant même par la création de textiles intelligents.

L'e-santé apparaît de plus en plus comme la solution à mettre en place pour palier aux difficultés de notre système de soins qui est confronté aujourd'hui à plusieurs défis majeurs :

- Le vieillissement de la population.
- La gestion de la dépendance.
- L'accès universel à une prise en charge de qualité.
- L'accroissement significatif des dépenses.
- L'explosion des maladies chroniques.
- L'évolution de la démographie médicale qui menace l'accès égalitaire aux soins.

Sur le papier, la e-santé semble être la solution alliant l'efficacité des soins apportés à la maîtrise des dépenses de santé mais sa généralisation implique de trouver des réponses à des questions de tous ordres telles que :

- La confidentialité des données personnelles.
- La gestion du déploiement des solutions techniques pour couvrir l'ensemble de la population mondiale.
- Le basculement vers le numérique des services de santé actuels.

La responsabilisation, la formation, l'autonomie, le suivi des patients lorsque les solutions d'e-santé leur permettront de rester à domicile pour leur traitement.

En effet, les technologies numériques, encore sous-utilisées dans ce secteur, représentent de véritables potentiels pour améliorer l'efficacité, la qualité et la sécurité de notre système de santé, le tout dans un contexte économique plus que tendu. Longtemps considéré comme un pays à la traîne en matière d'e-santé, la France ou plutôt le gouvernement semble avoir pris la mesure du chantier et de nombreux signes tangibles ont été visibles en 2011, concernant la mise en place d'une véritable politique de santé publique.

Il est à noter également qu'en 2011 (Vezin, 2012), le marché de l'e-santé grandit principalement en dehors de l'hôpital (plutôt orienté vers la future mise en place du DMP), au

plus près des patients dans leur lieu de domicile, avec deux cibles clairement identifiées qui sont :

- Les patients atteints de maladies chroniques (diabète, insuffisance cardiaque,...).
- Les séniors ou les personnes handicapées vivant à domicile et nécessitant une assistance spécifique.

11. Conclusion

Nous avons présenté à travers ce chapitre une vue globale sur l'architecture orientée service (SOA) : Les éléments de base d'une SOA, Les méthodes de développement et les différentes architectures.....etc. et la relation du SOA avec le domaine de la Santé.

Et enfin on a précisé les avantages et les inconvénients de cette approche, maintenant nous passons à une autre approche dans ce même domaine des services c'est l'approche orientée service décisionnelle (SOA+d), on va la présenter par la suite dans le prochain chapitre.

1. Introduction

Dans ce chapitre nous allons présenter la nouvelle méthodologie qui est dénommée SOA +d (Service Oriented Architecture with a Decision).

SOA +d s'adresse aux petites et moyennes entreprises pour lesquelles, il est indispensable de proposer une solution basée sur des standards. L'utilisation des standards constitue une boîte à outils à laquelle s'ajoute une démarche guidant la construction de l'architecture de services à chaque étape. En effet les standards UML (Unified Modeling Language), BPMN (Business Process Modeling Notation), DMN (Decision Model and Notation) et SoAML (Service Oriented Architecture Modeling Language) sont utilisés dans le cadre de la méthode SOA +d.

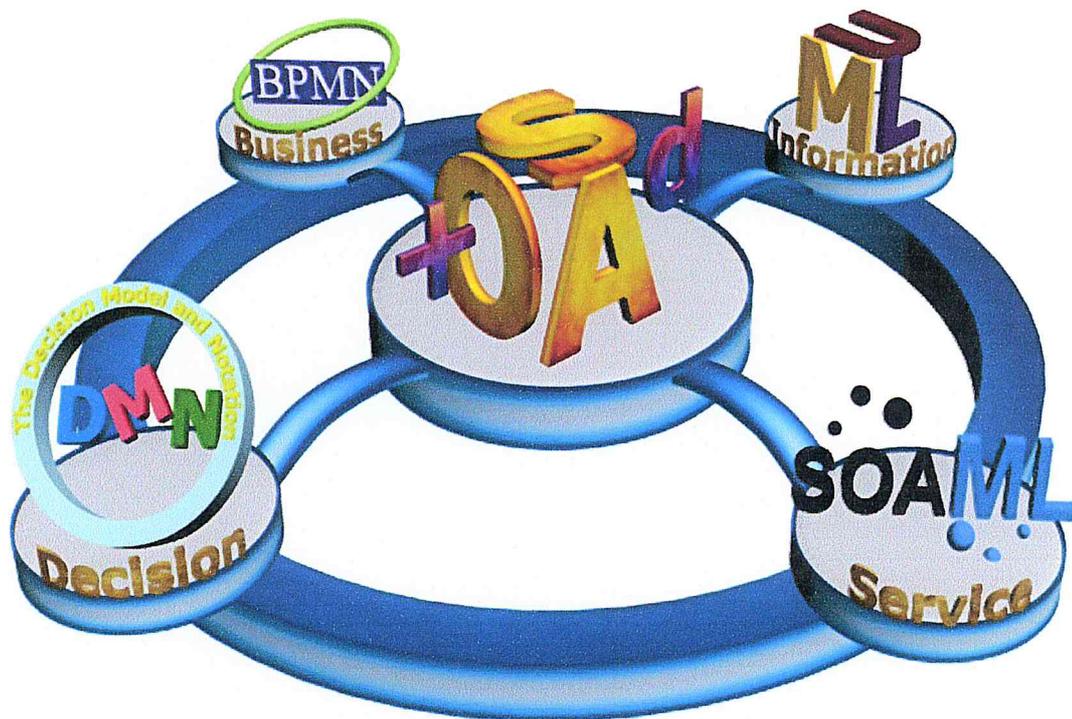


Figure II-1L'utilisation des standards par SOA+d (boumahdi, 2015).

2. Les langages de modélisation

2.1 Service Oriented Architecture Modeling Language (SoaML)

2.1.1 Définition de SoaML

SoaML (Service oriented architecture Modeling Language) est un langage de modélisation standard, dédié à la conception et à la modélisation de solutions SOA solutions utilisant le UML(Unified Modeling Language) .

SoaML est principalement défini comme un profil UML qui utilise les mécanismes d'extensions d'UML pour définir les concepts SOA en termes de concepts UML existants. SoaML peut être utilisé avec des outils UML "sur étagère" actuels, mais certains outils proposent des capacités de modélisation enrichies.

SoaML intègre des capacités de modélisation, afin de supporter l'usage de SOA à différents niveaux, et avec différentes méthodologies. En particulier, SoaML supporte les approches "contract-based" et "interface-based" qui, en général, suivent respectivement les éléments "ServiceContract" et "ServiceInterface" du profil SOA.

2.1.2 Le profil SoaML

Le profil SoaML est un modèle bien plus complexe, supportant une large gamme de méthodologies, et nécessitant une compréhension approfondie d'UML. Modeliosoft a déjà une implémentation d'un profil SoaML, publiée sous la forme d'un module open source dans le Modelio Store. L'utilisation de ce module SoaML ou du module TOGAF Architect doit être mutuellement exclusive, car chacun de ces deux modules met en oeuvre une approche différente à la modélisation SOA.

La spécification SoaML étend le méta-modèle UML avec un Profil UML qui adapte UML à la modélisation des services dans le cadre d'une architecture orientée services. Le profil UML SoaML fournit une syntaxe abstraite et trois syntaxes concrètes : une syntaxe concrète textuelle sous forme d'un fichier XMI (XML Metadata Interchange) et deux syntaxes concrètes graphiques avec deux formes de notations (les mots-clés et les icônes). Quant aux contraintes, elles sont décrites textuellement et ne sont ni formelles ni exprimées avec un langage de spécification de contraintes comme OCL.

Le **Tableau I-1** illustre quelques exemples de la syntaxe concrète graphique de SoaML.

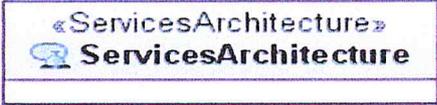
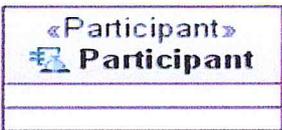
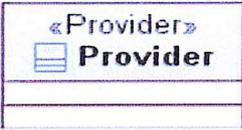
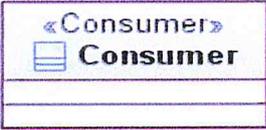
Mot clé	Icone	Exemple avec RSA
« <i>ServicesArchitecture</i> »		
« <i>ServiceContract</i> »		
« <i>Participant</i> »		
« <i>Provider</i> »	-	
« <i>Consumer</i> »	-	

Tableau I-2 Les deux syntaxes concrètes graphiques proposées par SoaML (Youness LEMRABET, 2012).

Le fait que la spécification SoaML ne propose pas des icônes pour tous les concepts (exemple : Provider et Consumer) prête à confusion. De plus, les exemples fournis dans la spécification sont illustrés avec deux outils différents : RSA (Rational Software Architect) et Cameo Enterprise architecture, alors que seul RSA utilise des icônes.

2.1.3 Présentation de SoaML

SoaML s'intéresse à l'architecture et non pas aux composants techniques sous-jacents. Son objectif est de fournir aux utilisateurs du langage UML, les moyens de modéliser une architecture orientée services comprenant des notions de clients et de fournisseurs de services, de collaboration de services ainsi que la notion de contrat. Dans cette section, nous présentons les différents concepts proposés par SoaML, dont le but de bien modéliser les architectures SOA.

❖ Les participants « Participants »

Ce concept est utilisé pour définir les fournisseurs et les consommateurs de services. Un participant peut représenter une personne, une organisation ou un système. Il peut jouer le rôle d'un fournisseur de services, d'un consommateur de services ou des deux en même temps. Le participant peut contenir des « *Service port* » s'il est fournisseur de services ou « *Request port* » s'il est consommateur de services.

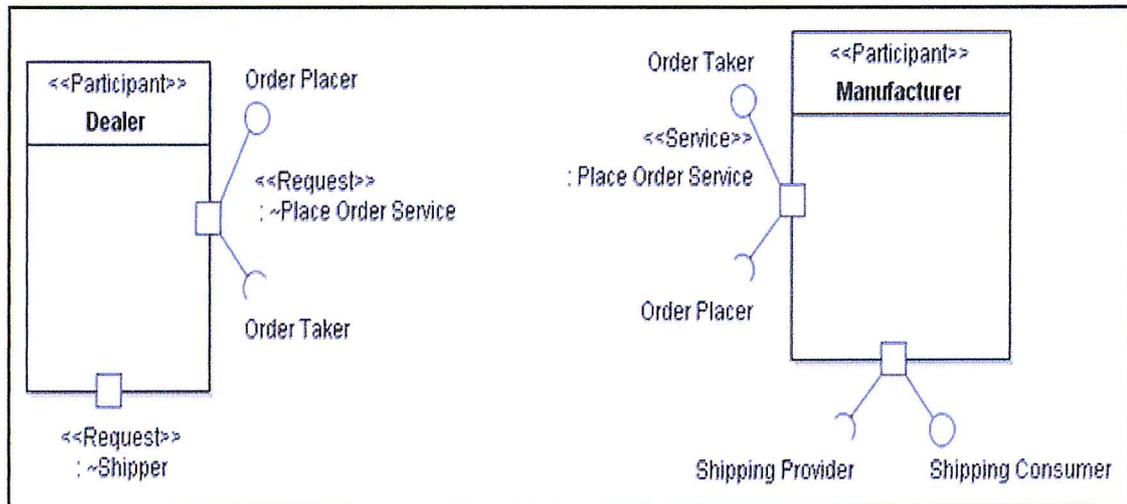


Figure II-2 Diagramme de participants (boumahdi, 2015).

❖ L'interface de service « ServiceInterface »

L'interface de service est utilisée pour décrire les opérations fournies ou requises pour compléter la fonctionnalité d'un service. Les interfaces des services aident à définir les interfaces et les responsabilités d'un Participant qui fournit ou utilise un service. Elles peuvent être utilisées pour typer les Service ports et les Request ports du Participant. Le *ServiceInterface* peut aussi spécifier le protocole d'utilisation des capacités d'un service avec les diagrammes de comportement UML (ex : diagrammes d'activité ou de séquence UML). Ceci permet de décrire le comportement des services et les échanges de messages entre les différentes parties. Enfin, le *ServiceInterface* n'exige pas la spécification du rôle du participant consommateur du service si ce dernier ne définit pas d'interface (i.e., interactions unidirectionnelle).

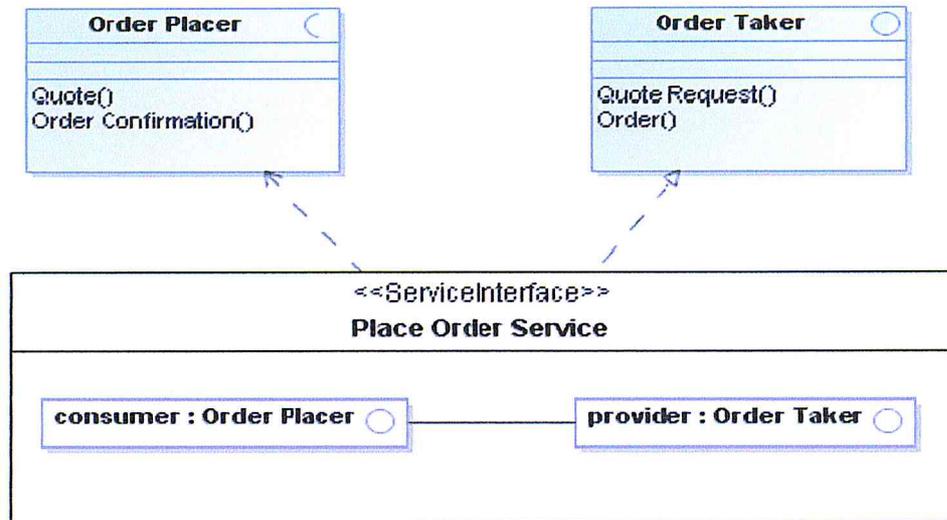


Figure II-3 Un diagramme d'interface de service (boumahdi, 2015).

❖ Le contrat de services « ServiceContracts »

Le contrat de services sert d'accord entre les différentes parties de la collaboration. Chaque rôle dans un ServiceContract est associé à une interface. Cependant, contrairement au Service Interface la spécification des rôles du fournisseur et du consommateur est obligatoire dans le *ServiceContract*. Ces rôles sont ensuite utilisés dans l'architecture des services pour relier les participants aux *CollaborationUse*. Enfin, le *ServiceContract* permet aussi de définir le comportement des services avec les diagrammes de comportements UML.



Figure II-4 Exemple de Contrat de service (boumahdi, 2015).

❖ L'architecture de services « Service Architecture »

L'architecture de services fournit une vue de haut niveau de l'architecture orientée service. Elle définit comment un ensemble de participants qui partagent un objectif commun collaborent en fournissant et en utilisant des services exprimés avec des contrats

de service (i.e., *ServiceContract*). L'architecture de services fournit le contexte dans lequel les participants collaborent, elle complète le point de vue des processus métiers qui décrivent le comportement des services (boumahdi,2015).

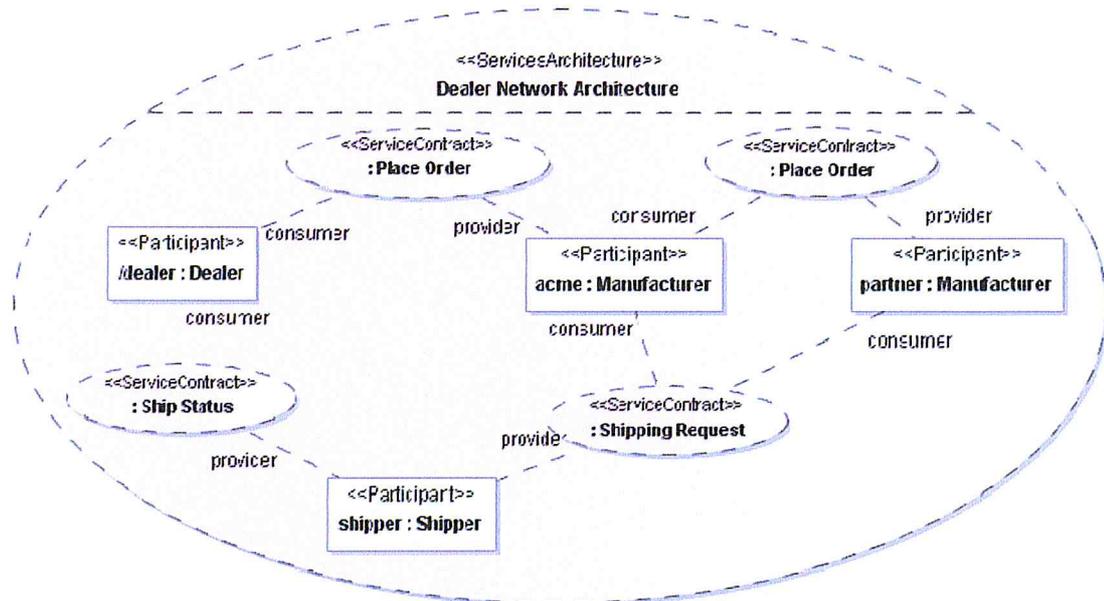


Figure II-5 Exemple d'architecture de services (boumahdi, 2015)

❖ Les données de services

Les données de services spécifient les informations échangées entre les consommateurs et les fournisseurs de services. Elles regroupent deux concepts « MessageType » et « Attachment ». Le premier est utilisé pour spécifier l'échange de données entre le consommateur et le fournisseur du service, alors que le deuxième représente une partie non contenue dans le message mais attachée à ce dernier. Il est à noter que MessageType ne contient aucune information spécifique à un protocole ou à une implémentation donnée. Enfin, ils existent deux styles de données échangées : le style orienté objet (Remote Procedure Call - RPC) et le style message ou document.

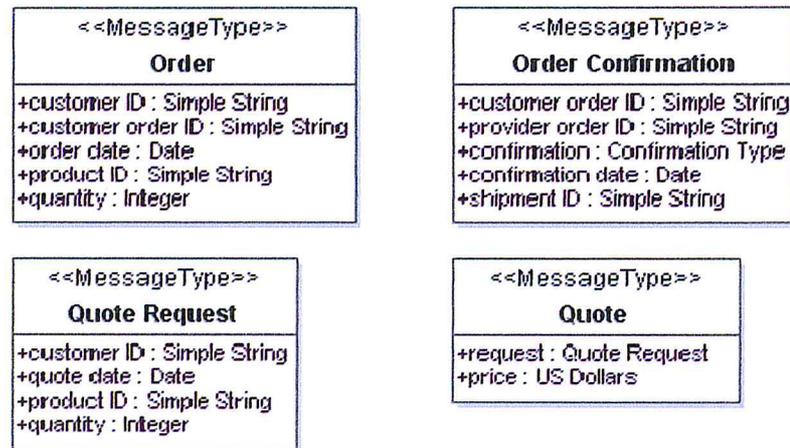


Figure II-6 Exemple de données de services (boumahdi, 2015).

❖ La Capacité « Capability »

Consiste à modéliser l’aptitude à agir et à produire un résultat qui peut être fourni par un service. La *Capability* peut être utilisée pour identifier les services requis dans le cadre de la collaboration. Elle peut aussi réaliser (au sens UML) des interfaces de services pour spécifier comment elles sont mises en œuvre par les participants.

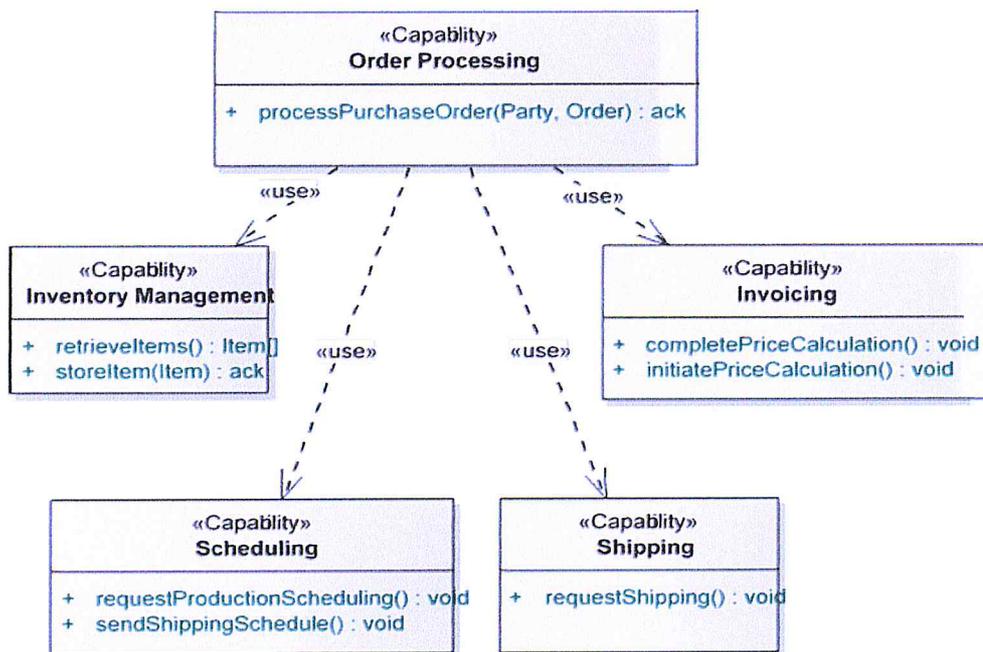


Figure II-7 Capacités de service nécessaires pour le traitement des commandes d’achat (boumahdi, 2015).

2.1.4 Les outils de SoaML

Nous pouvons utiliser les outils de modélisation des services pour modéliser la conception, la composition et les interactions entre services dans une architecture orientée service. Bien que SoaML soit un standard publié en Mars 2012, son support d'outil reste toujours limité. SoaML wiki de l'OMG [[omgwiki, 2011](#)] présente quelques outils de support disponibles pour SoaML. Lorsque cela est possible, nous préférons avoir un outillage open-source pour notre travail afin d'éviter d'être limité par les licences. En outre, nous préférons également des outils qui utilisent le métamodèle SoaML au lieu du profil UML.

2.2 Modèle de décision et de notation (DMN)

2.2.1 Définition de DMN

Le DMN (Modèle de décision et de notation) est un langage de modélisation de SOA, le but de ce langage est de normaliser les notations pour la modélisation des décisions.

2.2.2 L'utilisation de DMN

L'objectif principal de DMN est de fournir une notation commune facile à comprendre par tous les utilisateurs du métier, à partir des besoins des analystes métier pour créer les exigences de la décision initiale, puis des modèles de décision plus détaillées, pour les développeurs techniques chargés d'automatiser les décisions dans les processus, et enfin, pour les personnels de métier qui permettront de gérer et de contrôler ces décisions. DMN crée un pont standardisé pour l'écart entre la conception de la décision du métier et l'implémentation de la décision. La notation DMN est conçue pour être utilisable aux côtés de la notation des processus métier de la norme BPMN. La modélisation de décision est réalisée par des analystes du métier dans le but de comprendre et de définir les décisions utilisés dans une entreprise ou une organisation. Ces décisions sont généralement les décisions opérationnelles prises dans les processus métier au jour le jour, plutôt que la prise de décisions stratégiques pour lesquels il existe quelques règles et représentations.

Trois utilisations du DMN peuvent être distinguées dans ce contexte :

❖ Pour la modélisation de la prise de décision humaine

DMN peut être utilisé pour modéliser les décisions prises par le personnel au sein d'une organisation. La prise de décision humaine peut être décomposée en un graphe de décisions constitutives interdépendantes, et modélisée à l'aide d'un DRD. Les décisions du DRD peuvent probablement être décrites à un niveau assez élevé, en utilisant le langage naturel plutôt que la logique de décision.

❖ Pour la modélisation des exigences en matière de prise de décision

Automatisée

L'utilisation de DMN pour la modélisation des exigences en matière de prise de décision automatisée est similaire à son utilisation dans la modélisation de la prise de décision humaine, sauf qu'elle est entièrement normative plutôt que descriptive, et elle met davantage l'accent sur la logique de décision détaillée.

Pour l'automatisation complète des décisions, la logique de décision doit être complète, c'est à dire capable de fournir un résultat de décision pour chaque ensemble possible de valeurs des données d'entrée.

❖ Pour implémenter le processus décisionnel automatisé.

Si toutes les décisions et les modèles de connaissances métier sont entièrement spécifiés en utilisant la logique de décision, il devient possible d'exécuter des modèles de décision.

2.3 Business Process Model and Notation (BPMN)

2.3.1 *Definition BPMN*

Business Process Modeling Notation (BPMN) est une représentation graphique permettant de définir des processus métier dans un flux d'informations.

Business Process Modeling Notation (BPMN) est un standard pour la modélisation de processus métier qui fournit une notation graphique permettant de définir des processus métier dans un **diagramme de processus métier (BPM)**, basé sur une technique d'organigrammes très proche de celle utilisée par les diagrammes d'activité UML. L'objectif de BPMN est de supporter la **gestion des processus métier** pour les utilisateurs technique et métier, tout en étant capable de représenter des sémantiques complexes de processus. La

spécification BPMN fournit aussi une relation entre les graphiques de la notation et les concepts sous-jacents des langages d'exécution BPEL (**B**usiness **P**rocess **E**xecution **L**anguage).

2.3.2 Objectifs de BPMN

L'objectif principal de BPMN est de proposer une notation standard facilement compréhensible par les partenaires professionnels. Ces partenaires incluent les analystes métiers qui créent et raffinent les processus, les développeurs techniques responsables de l'implémentation des processus, et les directeurs commerciaux qui suivent et gèrent les processus. Par conséquent, BPMN est prévu pour servir comme langage visant à combler un déficit de communication qui survient souvent entre le design des processus métier et l'implémentation.

BPMN supporte seulement ces concepts de modélisation qui sont applicables aux processus métier. Ce qui signifie que d'autres types de modélisation réalisés par des organisations à caractères non commerciaux seront en dehors de la portée de BPMN. Par exemple, la modélisation de structures organisationnelles, les répartitions fonctionnelles et les modèles de données ne feront pas partie de BPMN.

3. Description globale de la méthode proposée SOA+d

La méthode que nous proposons dans ce chapitre rentre dans le cadre des approches descendantes. Les services identifiés par cette méthode auront une granularité allant de la granularité la plus faible à la granularité la plus élevée. Ceci permet d'avoir une architecture orientée services multicouches.

Les services ayant la granularité la plus faible sont les services les plus réutilisables. Ainsi dans une architecture SOA multicouches, les services techniques et les services d'intelligence doivent assurer un haut degré de réutilisabilité, contrairement aux services métier et services décision, pour lesquels la réutilisabilité n'est pas un facteur important.

Ces deux derniers types de services s'intéressent plutôt à la logique métier et la logique de décision encapsulée par les processus métier de l'entreprise.

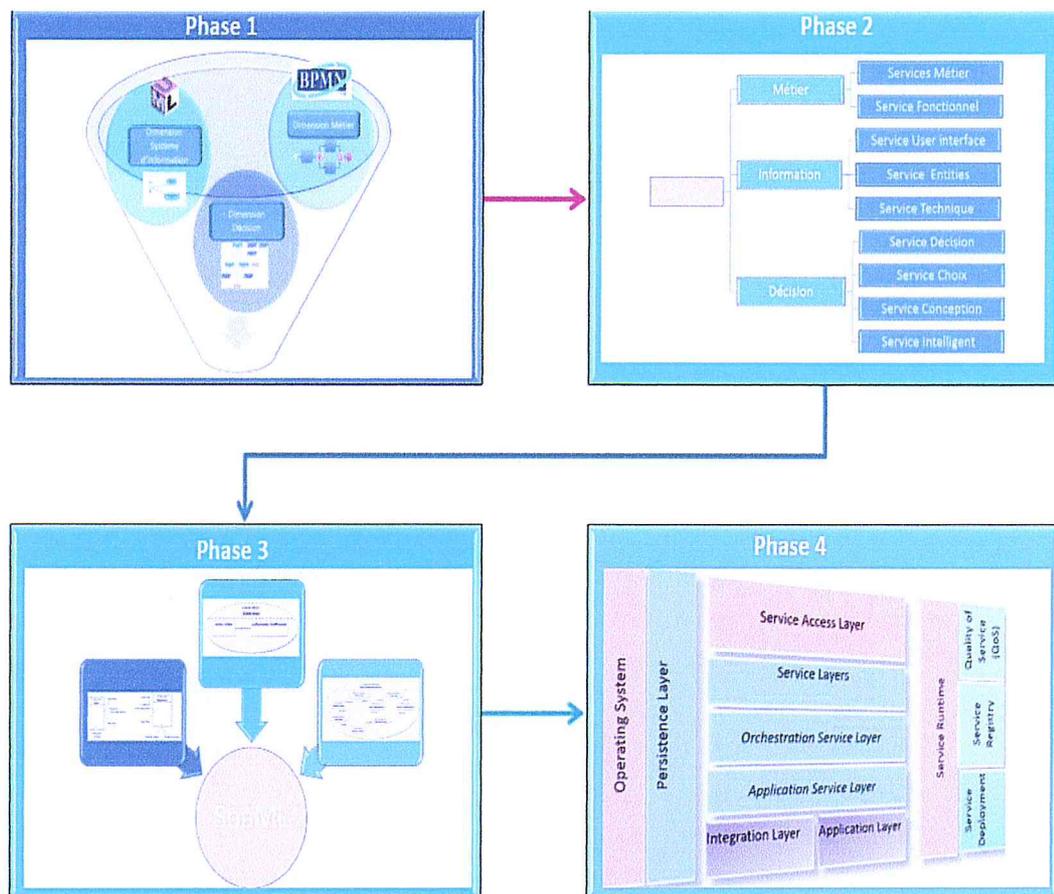


Figure II-8 Les phases de SOA+d (boumahdi, 2015).

4. Description détaillée de la méthode proposée SOA+d

SOA +d est une démarche méthodologique à quatre phases mettant en exergue une démarche de construction de l'architecture de services au sein de l'entreprise : la construction de la SOA métier, la construction de la SOA Informationnelle et la construction de la SOA Décisionnelle.

Chaque phase de la méthode SOA +d peut comporter une ou plusieurs étapes et peut être réalisée soit par un intervenant métier, un intervenant d'information, ou un intervenant de décision soit par les trois au même temps.

Nous allons présenter en détail les différentes phases de SOA+d en soulignant pour chaque phase l'ensemble des entrées-sortées et les intervenants impliqués.

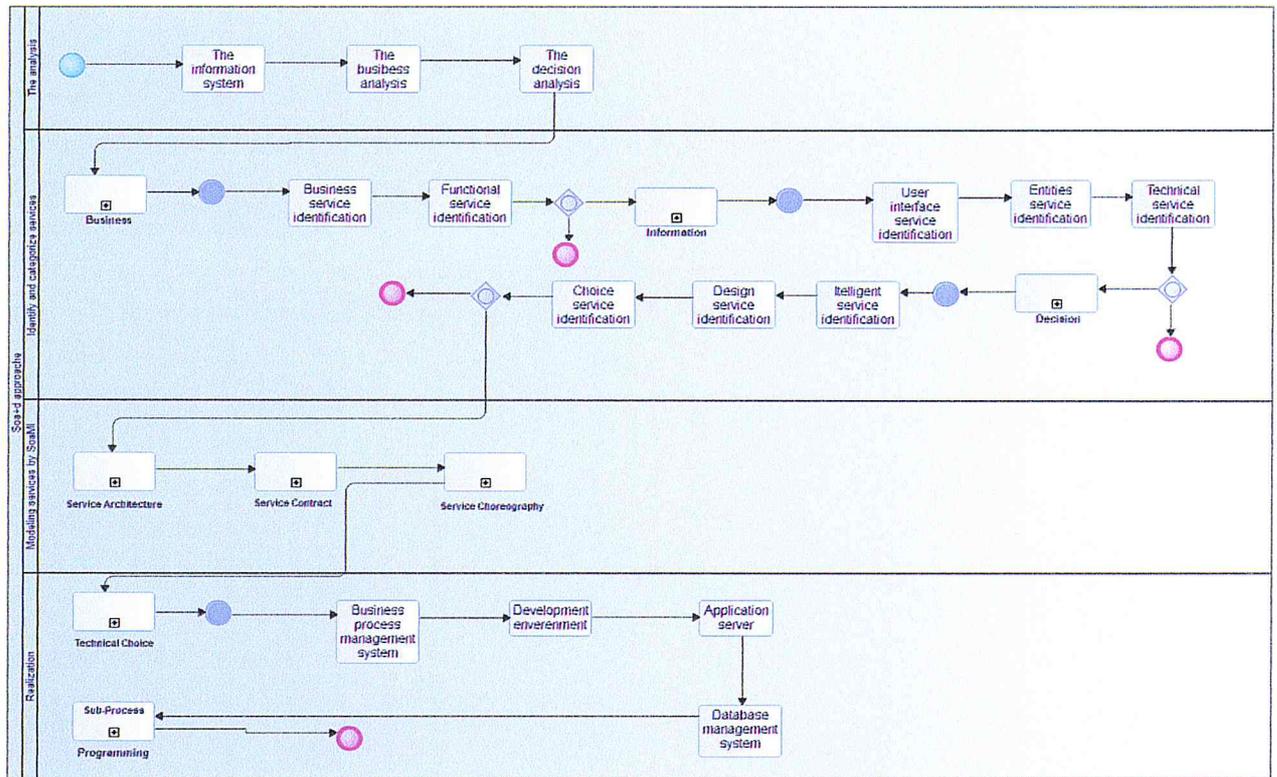


Figure II-9 La description des phases de SOA+d avec la notation BPMN (boumahdi, 2015).

4.1 Phase 1 : Analyse

La première phase de SOA +d est réalisée conjointement par les intervenants métier, les intervenants d’information et les décideurs de l’entreprise. Elle comporte trois étapes majeures : l’élaboration du modèle métier, l’étude du système d’information et la construction d’une ou des solutions aux problèmes décisionnels.

La phase d’analyse est définie en se basant sur un ensemble de modèles permettant de décrire les exigences métiers. Principalement, cette phase se compose des modèles des cas d’utilisation, BPMN et des notations DMN. Les modèles des cas d’utilisation permettent la formalisation des besoins des utilisateurs associés aux différents acteurs interagissant avec le système. Les modèles BPMN permettent la description des différents scénarios de réalisation des cas d’utilisations. A la fin, le DMN décrit la conception des solutions des décisions.

Cette phase comprend trois étapes: l’analyse du système d’information, l’analyse métier et l’analyse de décision.

4.1.1 Activité 1 : L'analyse du système d'information

L'analyse des systèmes d'information est une étude détaillée des différentes opérations effectuées par un système et leurs relations au sein et en dehors du système. Une question clé est : que faut-il faire pour résoudre le problème? Le but de l'analyse est de définir les limites du système et déterminer si un système candidat devrait envisager d'autres systèmes liés.

La première phase de notre processus de développement est l'élaboration des modèles des cas d'utilisation. L'objectif de cette phase est la capture et la spécification des besoins des utilisateurs ainsi que la description des processus métiers. Dans cette phase, SOA+d détermine, en premier lieu, les différents acteurs du système ainsi que les besoins de chaque acteur via les diagrammes des cas d'utilisations. En effet, Les diagrammes des cas d'utilisation jouent un rôle très important pour l'identification des besoins des utilisateurs. Ils décrivent de manière exhaustive les exigences fonctionnelles du système.

4.1.2 Activité 2 : L'analyse métier

Le modèle de processus métier est utilisé pour comprendre et décrire les processus métier, ce dernier est lié à l'environnement dans lequel le système à construire sera utilisé.

4.1.3 Activité 3 : L'analyse de décision

On définit trois étapes dans l'analyse de décision :

- Identifier les Décisions
- Spécifier les besoins de la décision
- Spécifier le niveau logique de la décision

4.2 Phase 2 : Identifier et catégoriser les services

Cette activité vise à identifier les services nécessaires à l'exécution du processus métier en cours de développement et c'est la première clé dans notre méthode. Les principales entrées de cette phase sont le BPMN, les cas d'utilisation et le DMN.

La découverte des services vise à apporter un soutien dans l'identification des services qui seront utilisés par les processus métier. Afin d'identifier et de décrire les services à partir des informations fournies par les cas d'utilisation, BPMN et le DMN. Nous proposons des règles de passage comme un guide pour identifier les services.

4.3 Phase 3 : Modélisation de Services

La phase 3 se concentre sur la spécification des services (modèle de données, interfaces, etc.). Dans un projet SOA, il est possible de se retrouver confronté à la problématique suivante : comment modéliser les services et leurs interactions? La notation SoaML offre une solution standardisée et basée sur l'UML et centrée sur la notion d'architecture.

Dans cette phase, les services doivent être modélisés avec un formalisme, nous adoptons une spécification à la base du langage SoaML qui offre un haut niveau d'abstraction, il est alors nécessaire d'affiner les services pour les rendre propres à une plate-forme donnée. SoaML permet d'axer la modélisation sur :

- L'identification des services, leurs exigences et leurs dépendances.
- La spécification d'un service : capacité, protocoles, politiques et informations transitant entre un service client et un service fournisseur.

À ce niveau le problème consiste à identifier les services de la solution SOA. Le but est d'identifier les services métiers publics à partir des exigences exprimées avec les processus. La solution la plus simple consiste à créer un seul service et d'y mettre toutes les opérations découvertes. Dans ce cas tous les participants deviennent dépendent de ce même service. Cette solution entraîne un couplage fort entre les participants. Tout changement par le fournisseur du service affectera l'ensemble des participants qui le consomment.

4.4 Phase 4 : La réalisation

Cette phase donne un aperçu des composants SOA et les options disponibles pour la mise en œuvre d'une architecture SOA à travers les Systèmes Open Source (SOS).

La **Figure I-10** illustre les différents choix technique qui doivent être pris en considération pour implémenter une architecture SOA. Dans les sections ci-dessous, les éléments d'architecture sont mappés au Système Open Source (SOS) outils/produits proposés.

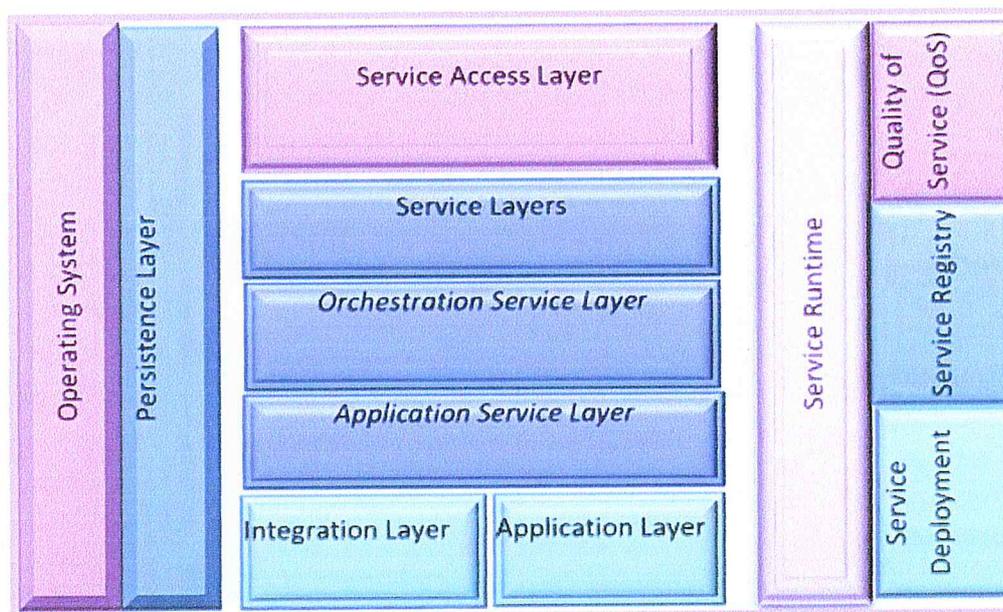


Figure II-10 Les choix technique pour implémenter une architecture SOA (boumahdi, 2015).

❖ Couche Accès Service

Est la couche interface entre le client final et le système. Nous pouvons utiliser le Serveur Apache HTTP pour implémenter cette couche.

❖ Couche Service

Cette couche définit et expose les fonctionnalités métier et application prises en charge par le système en tant que services. Sur la base de l'abstraction et de l'objectif, cette couche peut être encore divisée en :

- Couche service orchestration : Cette couche est fondamentale dans la réalisation de certains des principaux avantages de l'architecture SOA la composition, la réutilisation et l'abstraction. Les services de cette couche définissent et exposent les processus métier du système. Comme outil nous pouvons utiliser :
- jBPM de JBoss permet la création de processus métier qui coordonnent entre les personnes, les applications et les services.
- Le Moteur Active BPEL est capable d'exécuter des définitions de processus créés pour le Langage Business Process Execution Language.

❖ Couche Service Application

Les services de cette couche sont les «travailleurs». Ils représentent l'application logique réelle nécessaire pour mettre en œuvre les processus métier. Nous pouvons implémenter cette couche en utilisant Apache Axis, qui est une implémentation de SOAP et permet le développement et le déploiement de services.

❖ Couche Intégration

Cette couche intègre le système orienté services pour les systèmes existants. C'est important lorsque les systèmes existants ne peuvent pas être exposés en tant que services, qui peuvent interopérer directement avec le système orienté services. ServiceMix est un exemple de cette couche d'outils de mise en œuvre.

❖ Couche Application

Cette couche accueille les implémentations des services.

❖ Couche Persistance

Il s'agit généralement de la couche de base de données, où les données fonctionnelles et non fonctionnelles sont conservées temporairement (cache) ou d'une manière permanente (durée déterminée par les exigences métier et techniques). Pour implémenter cette couche, nous pouvons utiliser : Apache Derby, PostgreSQL, MySQL, ou Hibernate.

❖ Système d'exploitation

C'est le système d'exploitation sur lequel les différentes piles du système orienté services sont hébergés. Comme logiciel Open Source, nous pouvons utiliser Linux.

❖ Service Exécution

Cette pile représente les composants nécessaires pour déployer et «exécuter» les services. Les éléments de base de cette pile sont :

- **Service Regitre (Service Registry)** : Le registre est le référentiel central où tous les services sont définis, exposés en utilisant les spécifications standards, et découverts par les deux systèmes internes et externes.
- **Qualité de Service** : cette composante représente la fourniture de capacités de transactions sécurisées et fiables à des niveaux de performance optimaux.

- **Déploiement de Service:** Il s'agit de la plate-forme de déploiement de services. Cette plate-forme doit supporter les fonctionnalités d'une architecture orientée services.

5. Conclusion

Dans ce chapitre nous avons présenté la nouvelle méthode Soa+d, c'est une démarche top down, qui répond au manque de méthodes pour la construction de la SOA avec un aspect décisionnel. La démarche SOA +d a été décrite grâce à un processus de quatre phases qui guide les analystes dans le processus d'identification et de modélisation des services.

Dans le prochain chapitre nous allons utiliser cette méthode pour répondre aux problèmes de décision dans notre système de E-santé « diabète ».

Chapitre III: la conception

1. Introduction

Dans ce chapitre, nous allons présenter la démarche de développement de notre projet, en parcourant les différents diagrammes et services proposés par la méthode soa+d. Et à la fin nous terminons par la présentation de l'architecture technique de notre solution.

2. Phase 1 : Analyse

2.1 Analyse du système d'information

SOA+d propose l'utilisation du diagramme de cas d'utilisation pour modéliser l'aspect système d'information. La Figure suivante illustre notre diagramme de cas d'utilisation.

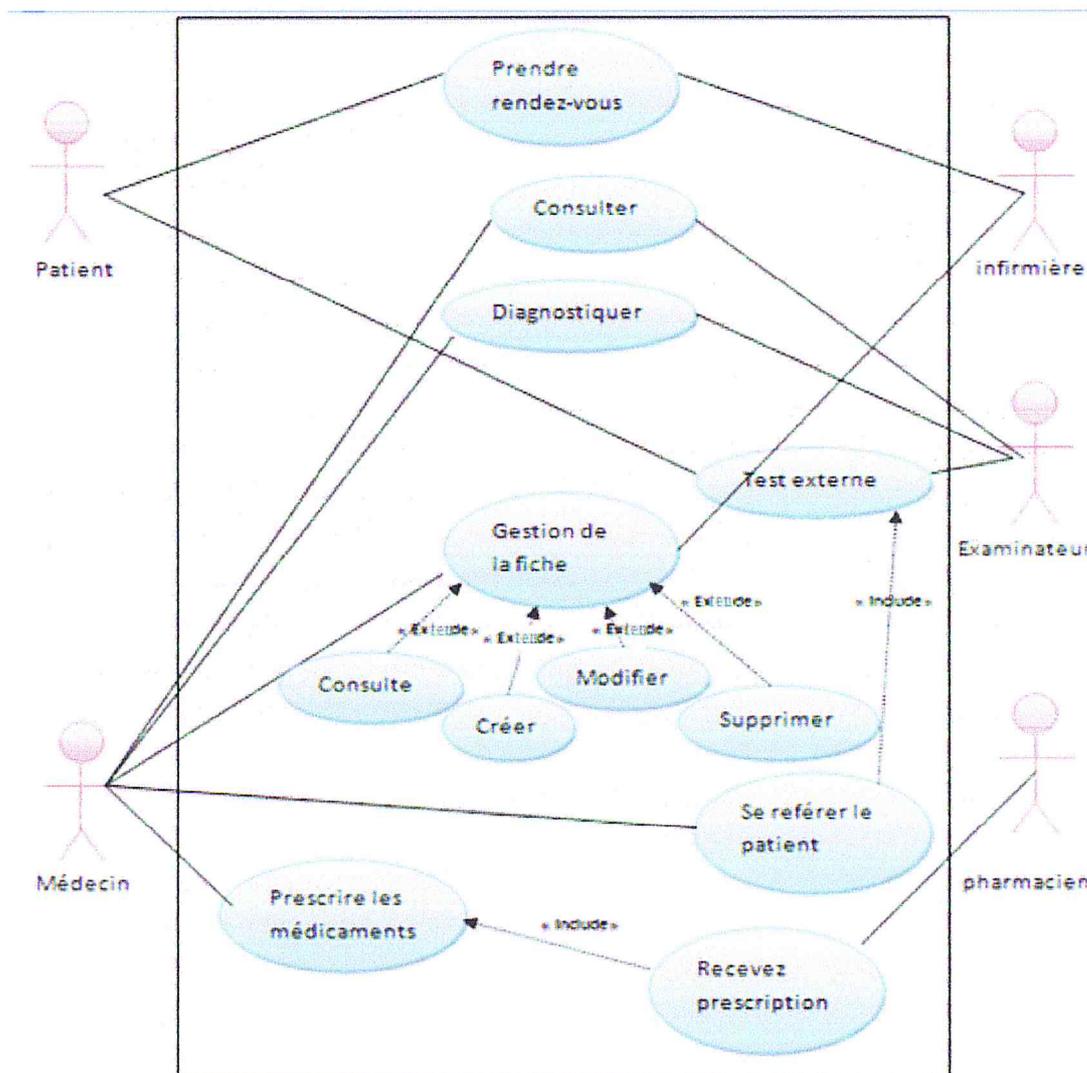


Figure I-1 Diagramme de cas d'utilisation.

2.2 Analyse métier

Nous utilisons le diagramme de BPMN pour modéliser l'aspect métier. Notre processus comporte deux sous processus:

Sous processus 1: Classement des patients

A l'arrivée du patient pour prendre un rendez vous, l'infirmière cherche ses informations: Dans le cas où il est déjà inscrit, elle vérifie ses informations, si sa nécessité une modification ou une suppression, elle met à jour la fiche et elle lui prépare le rendez vous sinon elle lui donne directement le rendez vous.

Sinon, si c'est un nouveau patient elle va l'inscrire et lui crée directement une nouvelle fiche et lui donne un rendez vous.

Le patient arrive au jour préciser par l'infirmière, Le médecin vérifie les informations écrites par l'infirmière dans la fiche et commence la consultation et les diagnostics nécessaires. Il cherche les symptômes du diabète insipide : vérifie si le patient souffre une polyurie et une polydipsie, si c'est le cas, il va lui faire un examen clinique, si le patient est en assez bon état ou bien les résultats d'examen prouvent que le patient est normal.

Dans ce cas le médecin confirme que le patient a un diabète insipide, mais il ne confirme rien au patient avant quelques examens et analyses complémentaires, si après ces derniers le médecin trouve que le pouls est augmenté, la tension artérielle diminuée, NA+sanguin est augmentée et le poids abaissé alors il peut confirmer au patient qu'il a un diabète insipide.

Sous processus 2: Classement des diabétique (Diabète insipide)

Après avoir confirmé que le patient a un diabète insipide, dans ce cas nous passons à ce sous processus. Dans ce cas le médecin met à jour la fiche et envoi une copie de cette fiche à l'examineur par un support informatique, et informe le patient qu'il doit connaître le type du diabète insipide pour faire le traitement, et que cette précision du type ne peut se faire que grâce à certains tests externes qui sont : le test d'ADH et le test de restriction hydrique.

Le jour où le patient visite l'examineur, ce dernier teste le patient et met les résultats sur la fiche reçu du patient et la renvoyé au médecin.

Quand le patient visite le médecin pour la prescription des médicaments, le médecin lui prescrit les médicaments selon le type du diabète insipide dérivé de l'analyse et envoie une copie de cette fiche au pharmacien, et le patient peut acheter les médicaments pour la guérison.

La **Figure I-2** montre la modélisation de notre processus avec la notation BPMN.

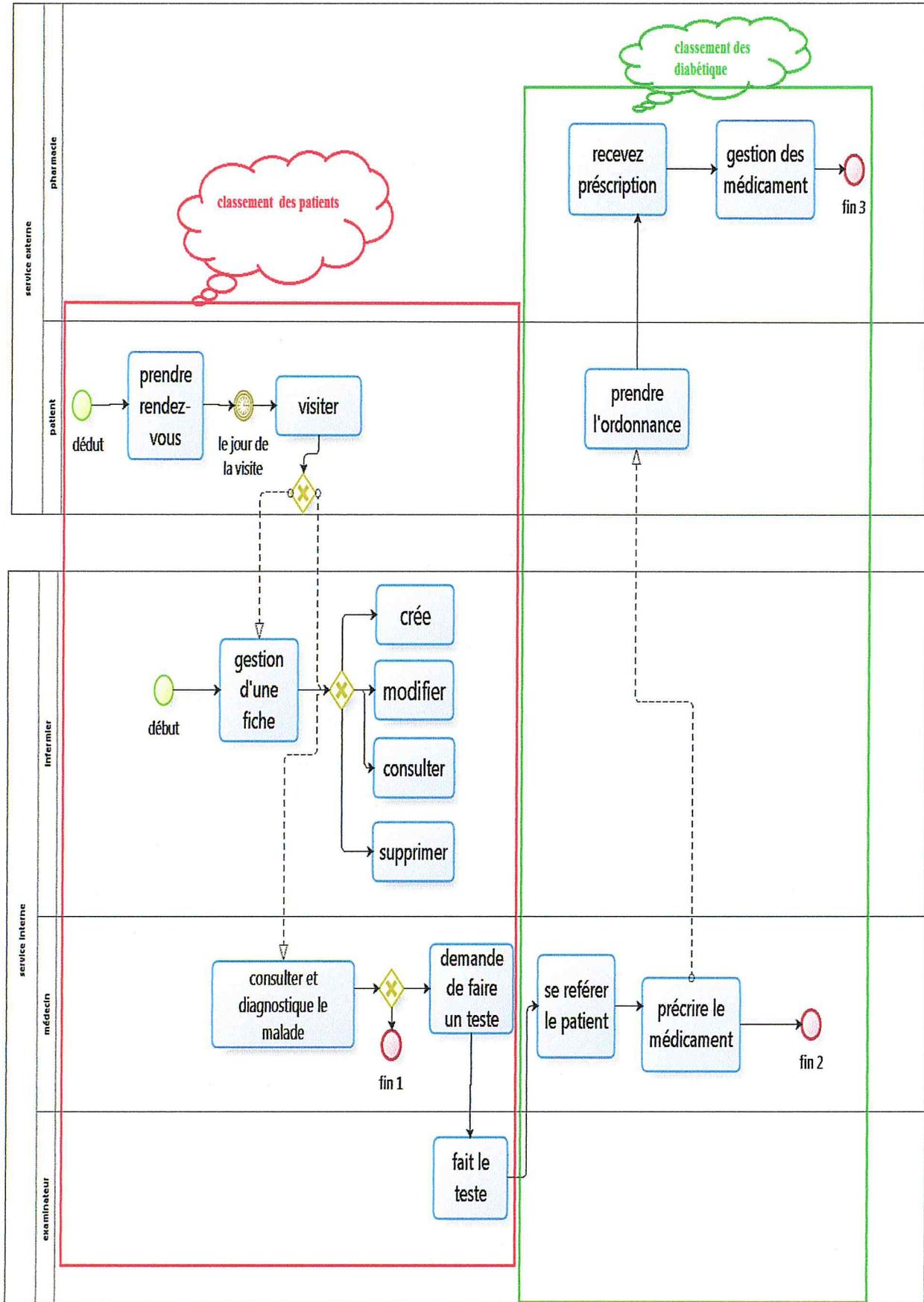


Figure I-2 Le diagramme BPMN de notre processus E-santé.

2.3 Analyse de la décision

D'après le cas d'utilisation et le BPMN, nous avons deux problèmes de décisions qui sont : le cas des patients et la précision du type du diabète insipide.

2.3.1 Le cas des patients

Le cas du patient comporte deux classes :

La classe des patients diabétiques (Diabète insipide) : ce classement se fait après l'apparition du patient en assez bon état général dans l'examen clinique et l'apparition de certains symptômes (une polyurie et une polydipsie...etc).

La classe des patients non diabétiques (Diabète insipide) : ce classement se fait après l'absence d'au moins l'un des symptômes du classement précédent.

La **Figure I-3** montre le DMN de la décision « classement des patients ».

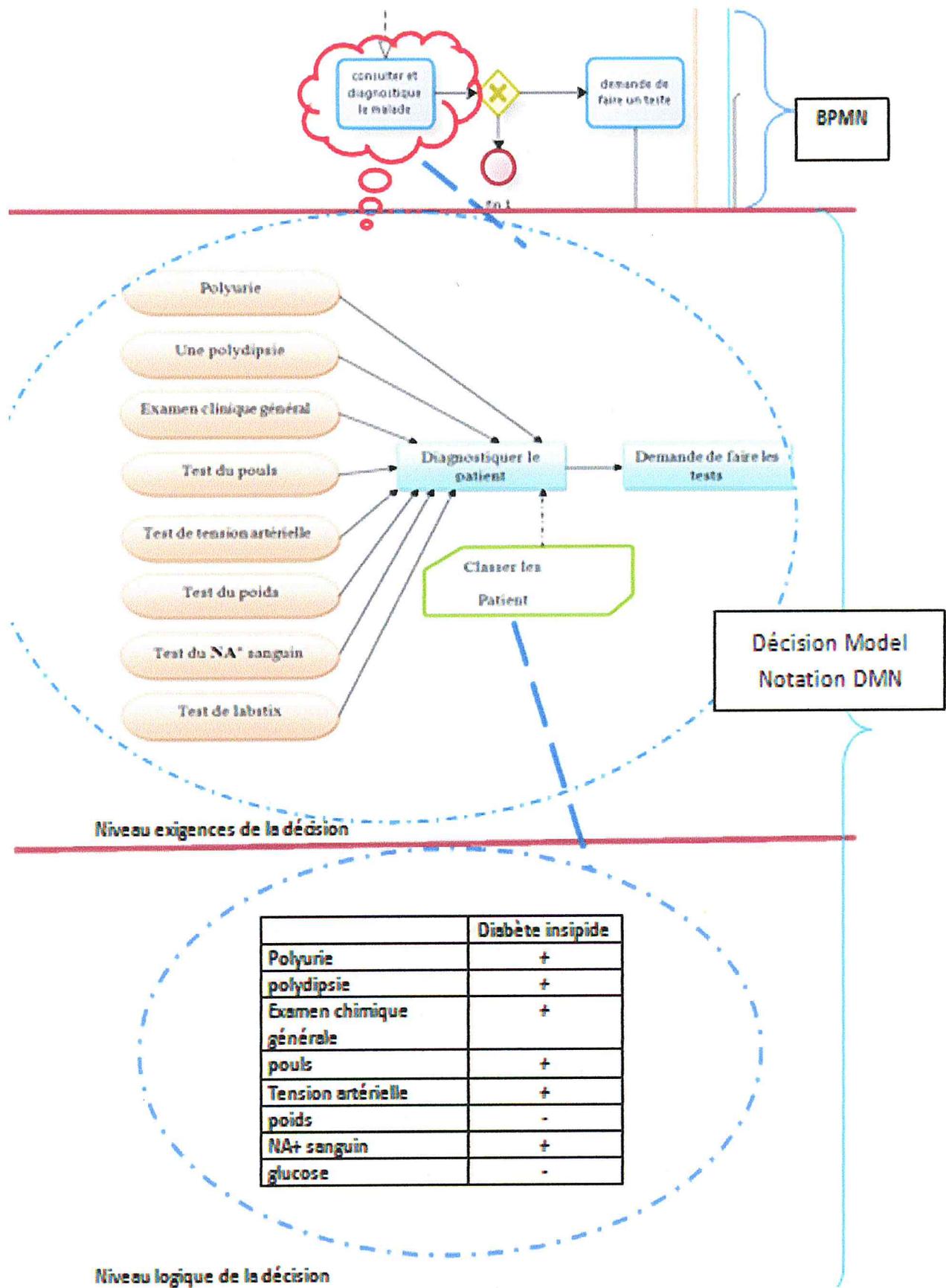


Figure I-3 Classement des patients DMN.

Le tableau suivant représente la matrice de décision destinée au problème de « classement des patients ».

	Diabète insipide
Polyurie	+
Polydipsie	+
Examen clinique générale	+
Pouls	+
Tension artérielle	+
Poids	-
NA+ sanguin	+
Glucose	-

Afin de montrer la méthode de prise de décision pour le classement d'un patient, nous allons présenter l'explication et la représentation logique de ce problème.

➤ **Explication logique**

POLYURIE_{pos} : augmentation du volume des urines.

POLYDIPSIE_{pos} : le patient boit sans cesse.

EXAM_{nor} : l'Examen clinique est normal.

POUL_{pos} : le pouls est augmenté.

TENTIN_{neg} : tension artérielle diminuée.

POID_{neg} : poids abaissé.

NA_{pos} : NA⁺ sanguin est augmentée.

GLYCEMIEnor : la quantité de la glycémie est dans état normal.

GLUCOSEneg : pas de glucose dans labstisc.

X : le patient.

La figure suivante illustre la représentation logique de la prise de décision 'Classement des patients' par les formules logiques du premier ordre :

$$\begin{aligned} & \mathbf{x/POLYURIEpos(x)^POLYDIPSIEpos(x)^EXAMnor(x)^} \\ & \mathbf{POULpos(x)^TENTINneg(x)^POIDneg(x)^NApos(x)^GLY} \\ & \mathbf{CEMIEnor(x) ^GLUCOSEneg(x) \Rightarrow Diabète insipide(x)} \end{aligned}$$

Figure I-4 La représentation logique de la prise de décision 'Classement des patients'

2.3.2. La précision du type du diabète insipide

La précision du type du diabète insipide comporte trois classes (trois types de diabètes insipides) :

Diabète insipide central par carence totale : l'appartenance à cette classe se fait après un résultat positif (+) du test ADH et un résultat positif (+) du test de la restriction hydrique.

Diabète insipide central par carence partielle : l'appartenance à cette classe se fait après un résultat positif (+) du test ADH et un résultat négatif (-) du test de la restriction hydrique.

Diabète insipide central par carence totale : l'appartenance à cette classe se fait après un résultat négatif (-) du test ADH et un résultat négatif (-) du test de la restriction hydrique.

La **Figure I-5** montre le diagramme DMN de la décision « Le classement des malades selon leurs types du diabète insipide ».

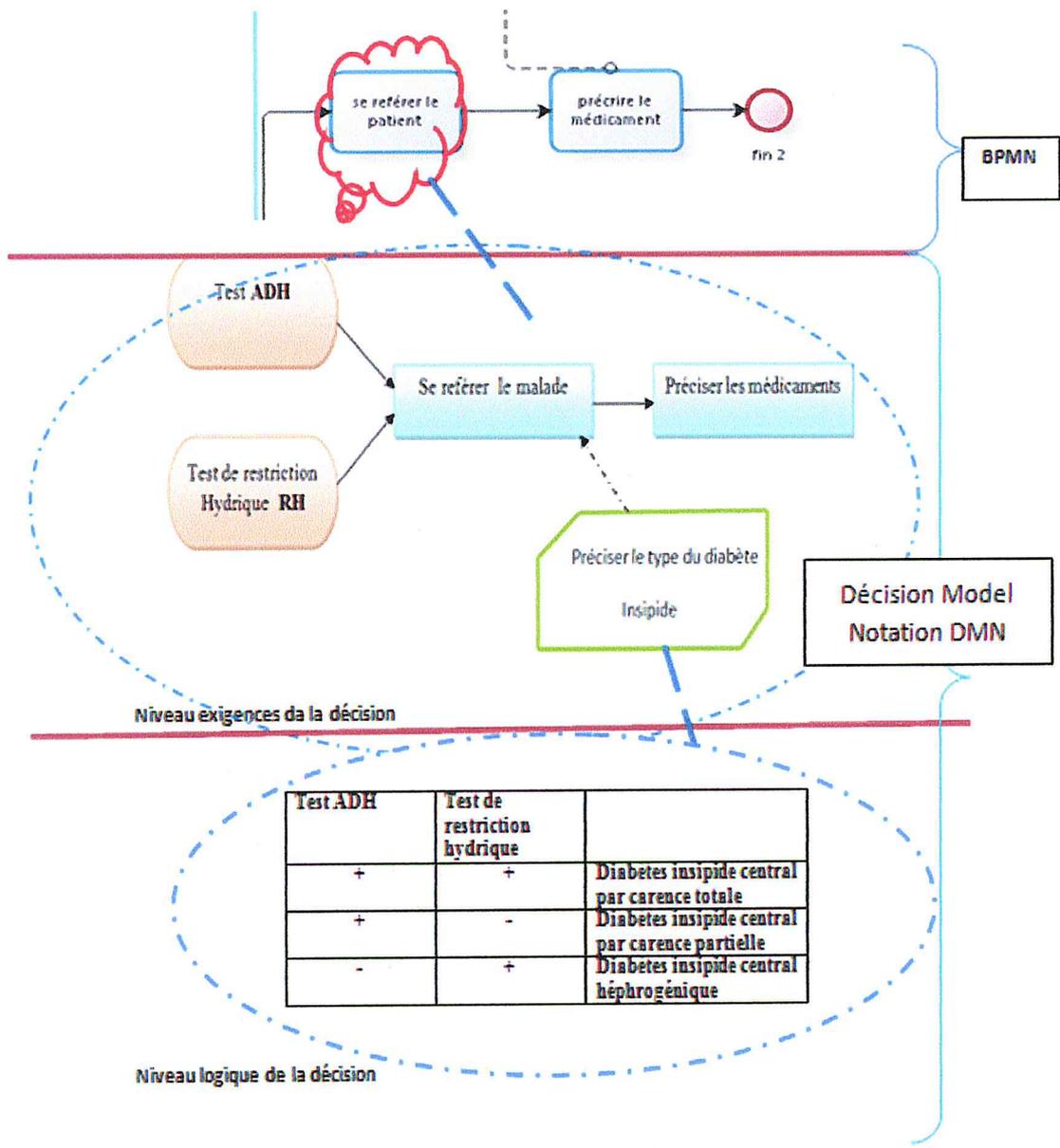


Figure I-5 Le classement des malades selon leurs types du diabète insipide DMN.

Le tableau suivant illustre la matrice de décision utilisée pour résoudre le problème de décision « classement diabétique ».

Test ADH	Test de restriction hydrique	
+	+	Diabète insipide central par carence totale
+	-	Diabète insipide central par carence partielle
-	+	Diabète insipide central néphrogénique

Pour comprendre la méthode de prise de décision nous avons opté à une représentation logique de notre problème.

➤ **Explication logique**

ADHpos : le résultat du test d'ADH est positif.

RHpos : le résultat du test de la restriction hydrique est positif.

ADHneg : le résultat du test d'ADH est négatif.

RHneg : le résultat du test de la restriction hydrique est négatif.

X : patient.

La figure suivante illustre représentation logique de la prise de décision 'Classement diabétique' par les formules logiques du premier ordre :

$x/ADHpos(x) \wedge RHpos(x) \Rightarrow \text{Diabète insipide central par carence totale}(x)$

$x/ADHpos(x) \wedge RHneg(x) \Rightarrow \text{Diabète insipide central par carence partielle}(x)$

$x/ADHneg(x) \wedge RHpos(x) \Rightarrow \text{Diabète insipide central néphrogénique}(x)$

Figure I-6 La représentation logique de la prise de décision 'Classement diabétique'.

3. Phase 2 : Identifier et classer les services

Dans cette phase nous identifions les trois types des services suivants : services qui existent au niveau métier, système d’information et les services décisionnels.

La découverte des services vise à apporter un soutien dans l’identification des services qui seront utilisés par les processus métier. Afin d’identifier et de décrire les services à partir des informations fournies par les cas d’utilisation, BPMN et le DMN, nous utilisons Les règles proposer par (Boumahdi, 2015) .

Règle de passage 1 : Cas d’utilisation Global vers le service métier

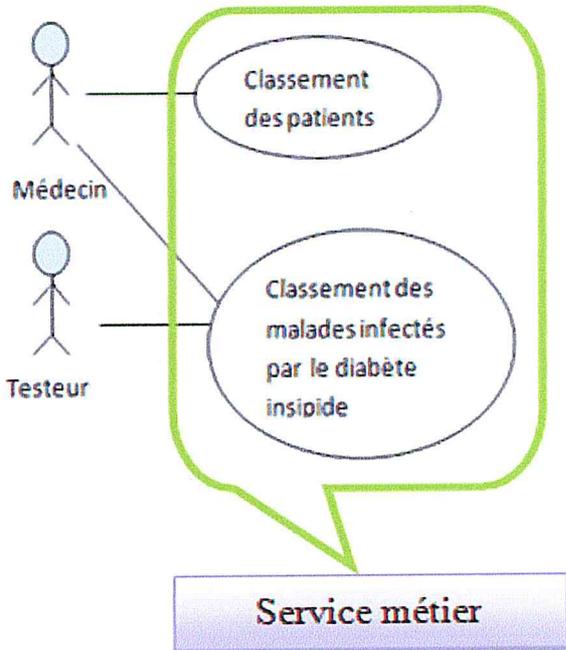
Cas d’utilisation	Service	Applique sur notre problème
Cas d’utilisation global	Métier	

Tableau I-1 Règle de passage 1.

Règle de passage 2 : Cas d'utilisation CRUD vers le service CRUD

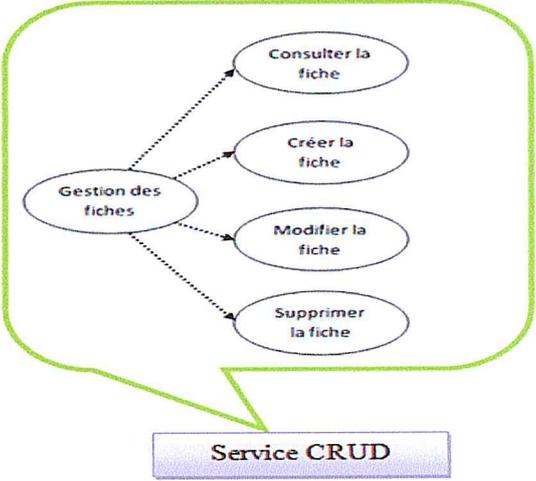
Cas d'utilisation	Service	Applique sur notre problème
Cas d'utilisation CRUD	CRUD	

Tableau I-2 Règle de passage 2.

Règle de passage 3 : Activité de BPMN vers le service Fonctionnel

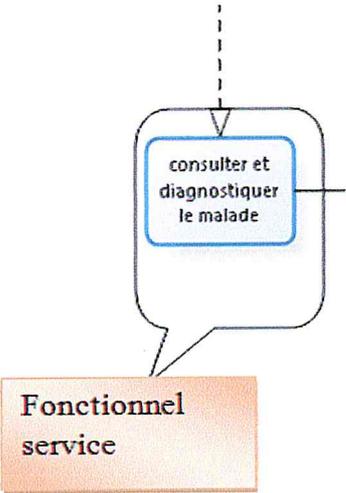
BPMN	Service	Applique sur notre problème
Activité	fonctionnel	

Tableau I-3 Règle de passage 3.

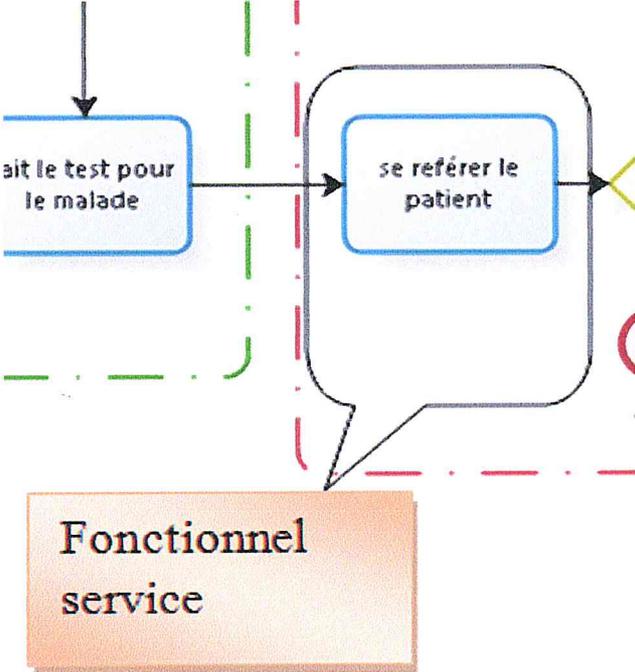
BPMN	Service	Applique sur notre problème
Activité	fonctionnel	 <p>The diagram illustrates the mapping of a BPMN activity to a functional service. On the left, a BPMN activity box labeled 'ait le test pour le malade' has an incoming arrow from above. An arrow points from this activity to a rounded rectangular box labeled 'se référer le patient'. This box is enclosed within a larger rounded rectangular frame representing the functional service. Below this frame is a callout box labeled 'Fonctionnel service'. Dashed lines in green and red are present in the background, and a yellow arrow points out from the right side of the 'se référer le patient' box.</p>

Tableau I-4 Règle de passage 3.

Règle de passage 4 : Tâche de BPMN vers service Technique

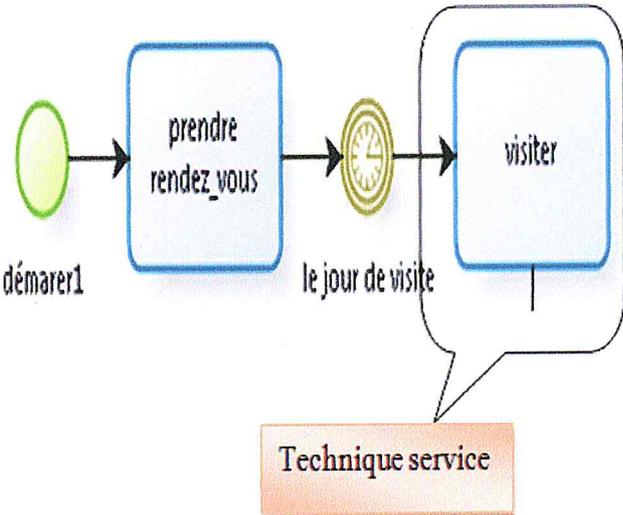
BPMN	Service	Applique sur notre problème
Tâche	technique	 <p>The diagram illustrates the mapping of a BPMN task to a technical service. On the left, a BPMN task box labeled 'prendre rendez_vous' is preceded by a green oval labeled 'démarrer1'. An arrow points from 'démarrer1' to the task box. Another arrow points from the task box to a yellow clock icon labeled 'le jour de visite'. A final arrow points from the clock icon to a rounded rectangular box labeled 'visiter'. This box is enclosed within a larger rounded rectangular frame representing the technical service. Below this frame is a callout box labeled 'Technique service'.</p>

Tableau I-5 Règle de passage 4.

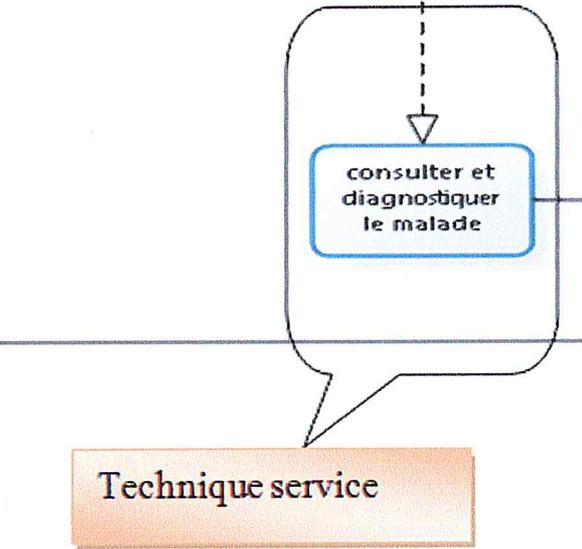
BPMN	Service	Applique sur notre problème
Tâche	technique	

Tableau I-6 Règle de passage 4.

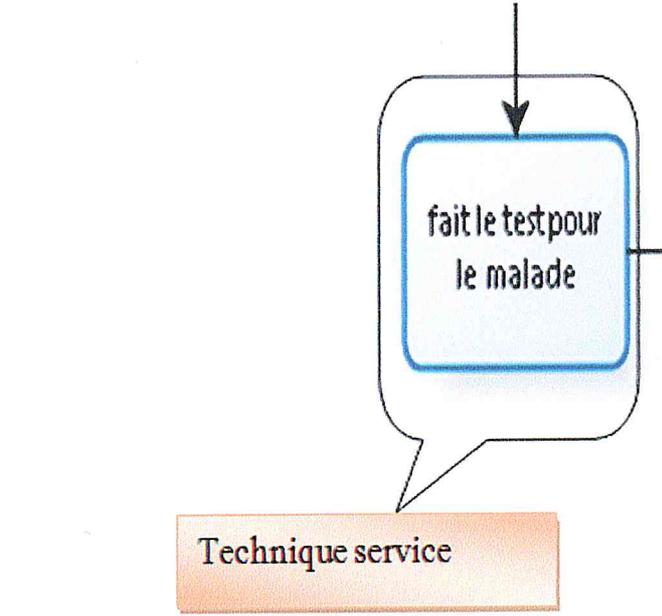
BPMN	Service	Applique sur notre problème
Tâche	technique	

Tableau I-7 Règle de passage 4.

BPMN	Service	Applique sur notre problème
Tâche	technique	

Tableau I-8 Règle de passage 4.

Règle de passage 5 : Indice de Décision vers Service Décision

Cas d'utilisation +BPMN	Service	Applique sur notre problème
Indice de décision	Décision	

Tableau I-9 Règle de passage 5.

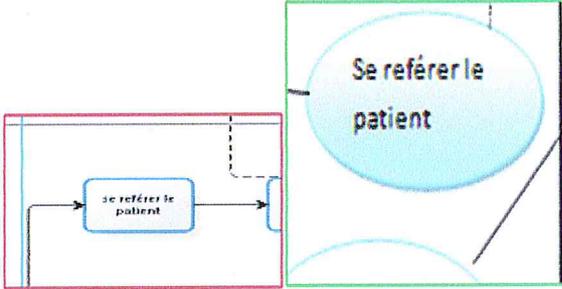
Cas d'utilisation +BPMN	Service	Applique sur notre problème
Indice de décision	Décision	

Tableau I-10 Règle de passage 5.

Règle de passage 6 : Décisions de DMN vers Décision de service

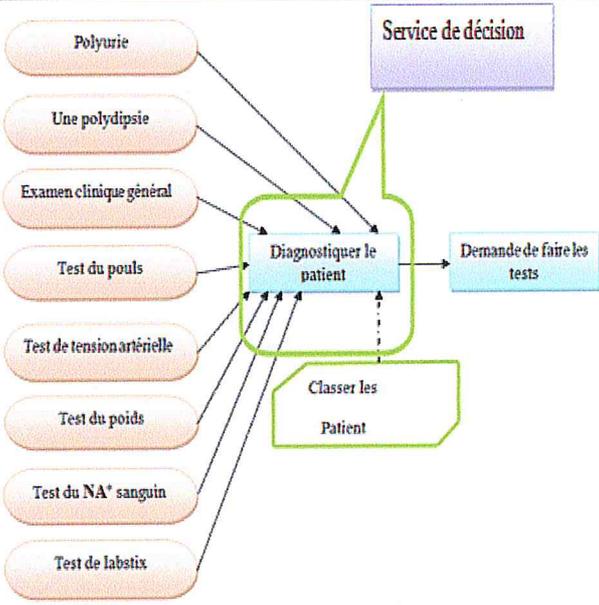
DMN	Service	Applique sur notre problème
Décisions	Décision	

Tableau I-11 Règle de passage 6.

DMN	Service	Applique sur notre problème
Décisions	Décision	

Tableau I-12 Règle de passage 6.

Règle de passage 7 : Connaissance Métier DMN vers service de conception

DMN	Service	Applique sur notre problème
Connaissance métier	Conception	

Tableau I-13 Règle de passage 7.

DMN	Service	Applique sur notre problème
Connaissance métier	Conception	

Tableau I-14 Règle de passage 7.

Règle de passage 9 : Données en entrée du DMN vers Service Intelligent

DMN	Service	Applique sur notre problème
Donnée en entrée	Intelligent	

Tableau I-15 Règle de passage 9.

DMN	Service	Applique sur notre problème
Donnée en entrée	Intelligent	

Tableau I-16 Règle de passage 9.

La Figure I-2 résume les services identifiés dans chaque vision (métier SI et décision).

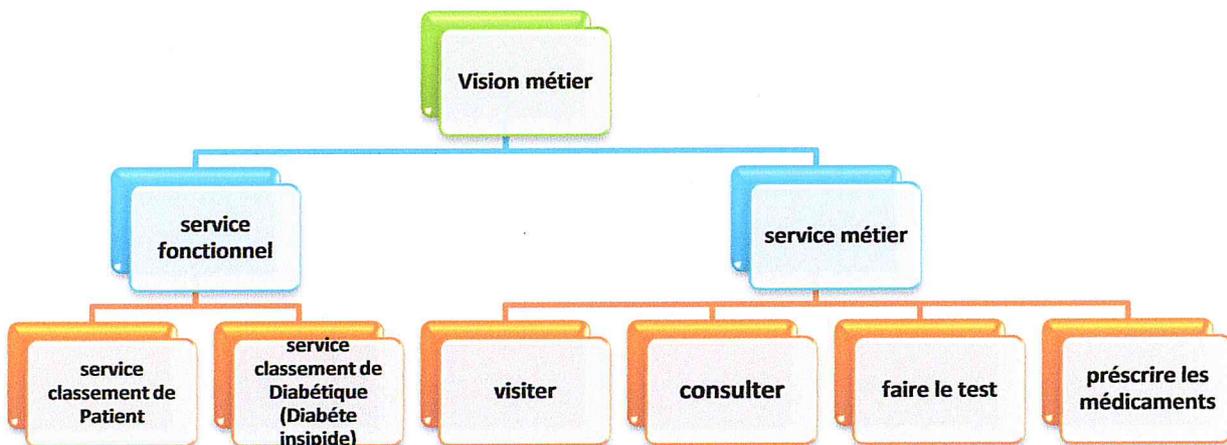


Figure I-7 Les services de la vision Métier.

Après avoir détaillé le diagramme de cas d'utilisation, et appliquer les règles de passage (Boumahdi, 2015), nous avons identifié les services de la dimension « Système d'information », qui sont présentés dans la Figure suivante :

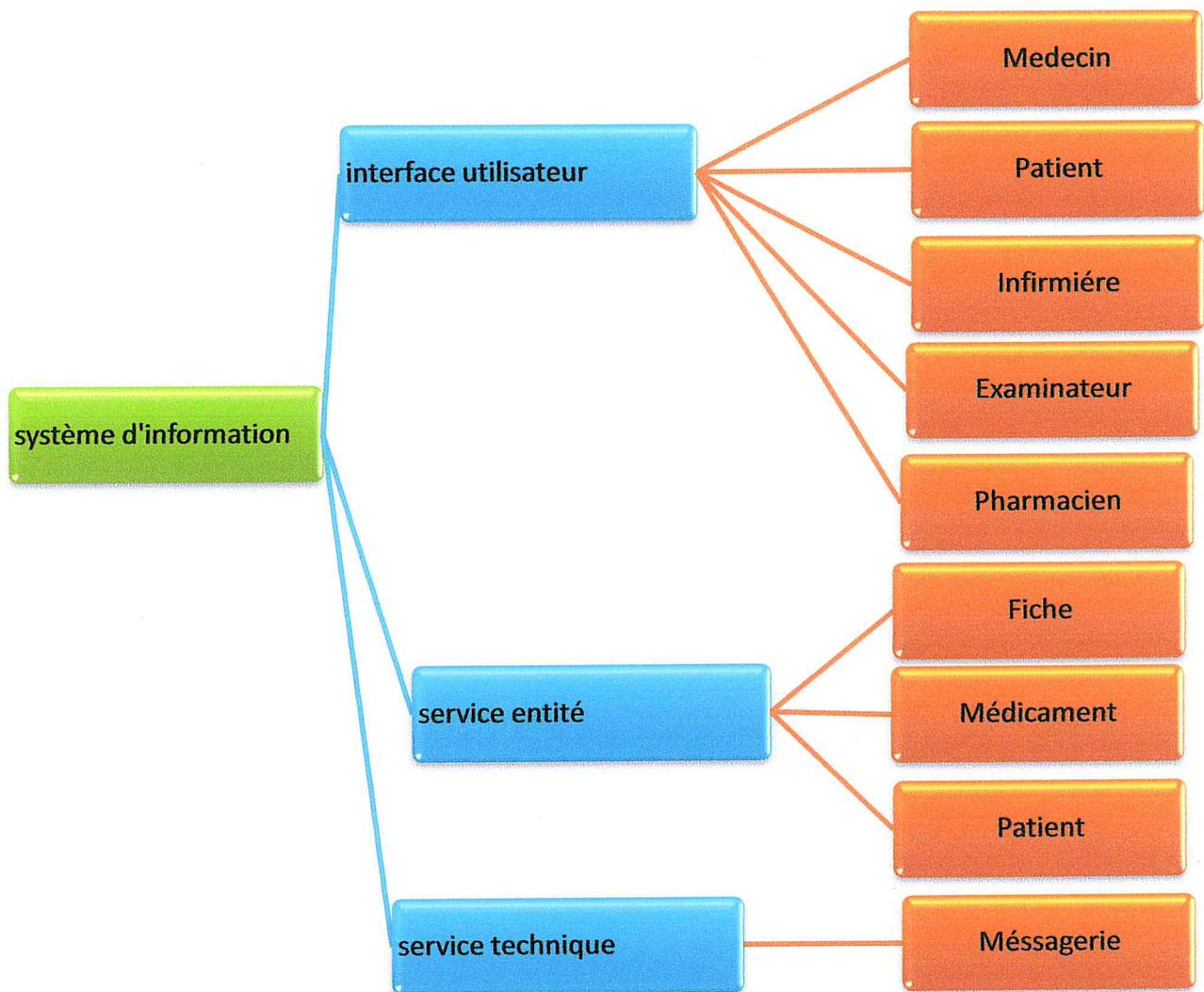


Figure I-8 Les services de la vision «Système Information».

Afin de répondre aux problèmes de décision identifiés dans la Phase 1, un ensemble de services est construit à la base des règles de passage. La vision décision est présentée dans la Figure suivante :

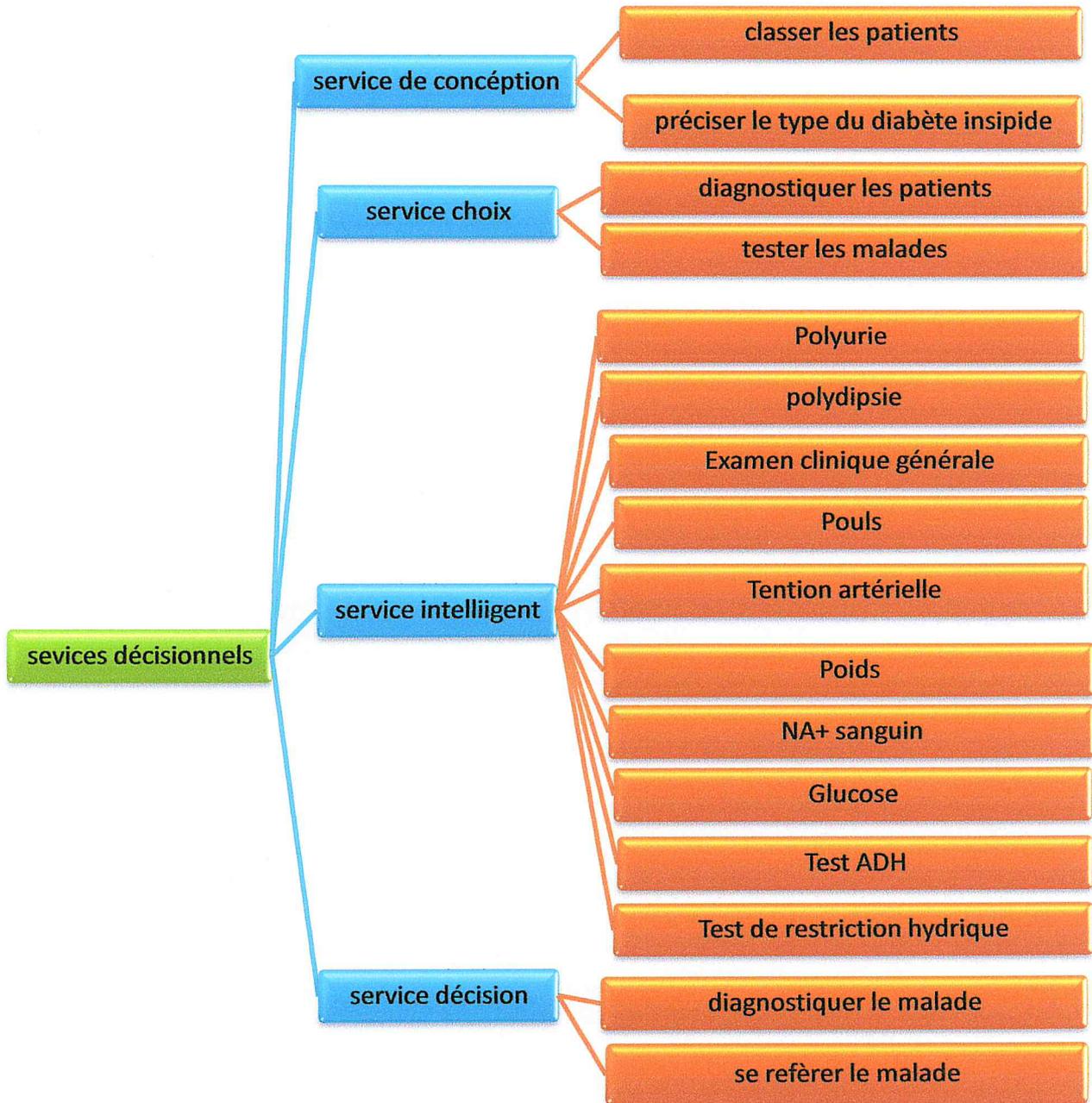


Figure I-9 Les services de la vision «Décision».

4. Phase 3: Modélisation des services

Pour passer d'un modèle BPMN à un modèle SoaML, il faut respecter un ensemble de règles proposées par (Brian Elvesæter, 2010), dont le but est de présenter la nouvelle version du langage de modélisation de l'architecture SOA, SoaML.

4.1 SoaML

En proposant SoaML (Service Oriented Architecture Modeling Language) l'OMG a adapté UML qui est fortement imprégné des principes de l'orienté objet aux principes de l'orienté service (LEMRAËT, 2012).

La spécification SoaML définit un langage de modélisation des services au sens SOA.

SoaML est avant tout destiné à modéliser des services dans des environnements distribués en prenant en compte les deux vues : métiers et IT. Le profil SoML permet de personnaliser les abstractions UML nécessaires à la modélisation des services. Il clarifie comment utiliser les métaclasses UML pour modéliser une solution SOA afin de garantir que les modèles des services reflètent les cinq principes SOA suivants : couplage faible, abstraction, réutilisation, autonomie et composition (LEMRAËT, 2012).

La **Figure I-10** présente le diagramme " Architecture générale" de SOAML pour notre cas d'étude "E-santé".

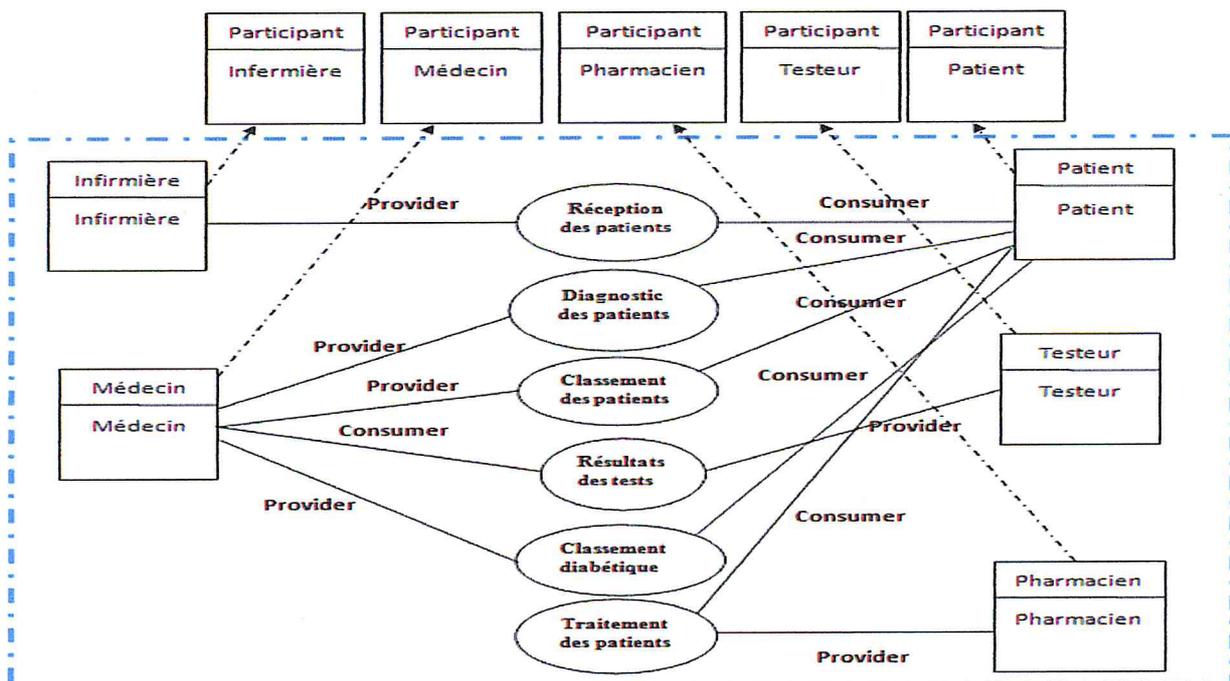


Figure I-10 Architecture générale.

Les Figures I-11 jusqu'à la figure I-16 illustrent les diagrammes "contrat de service" de SOAML liés à notre processus "E-santé".

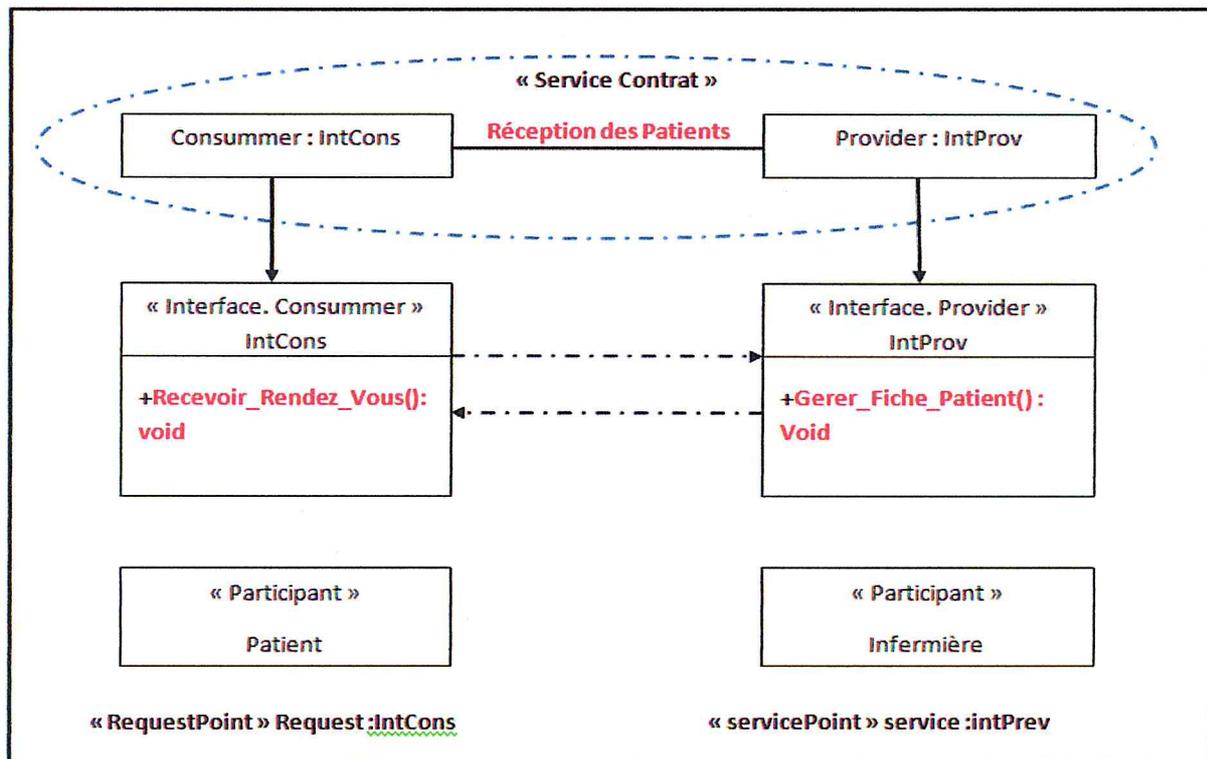


Figure I-11 Le contrat de service Réception des patients.

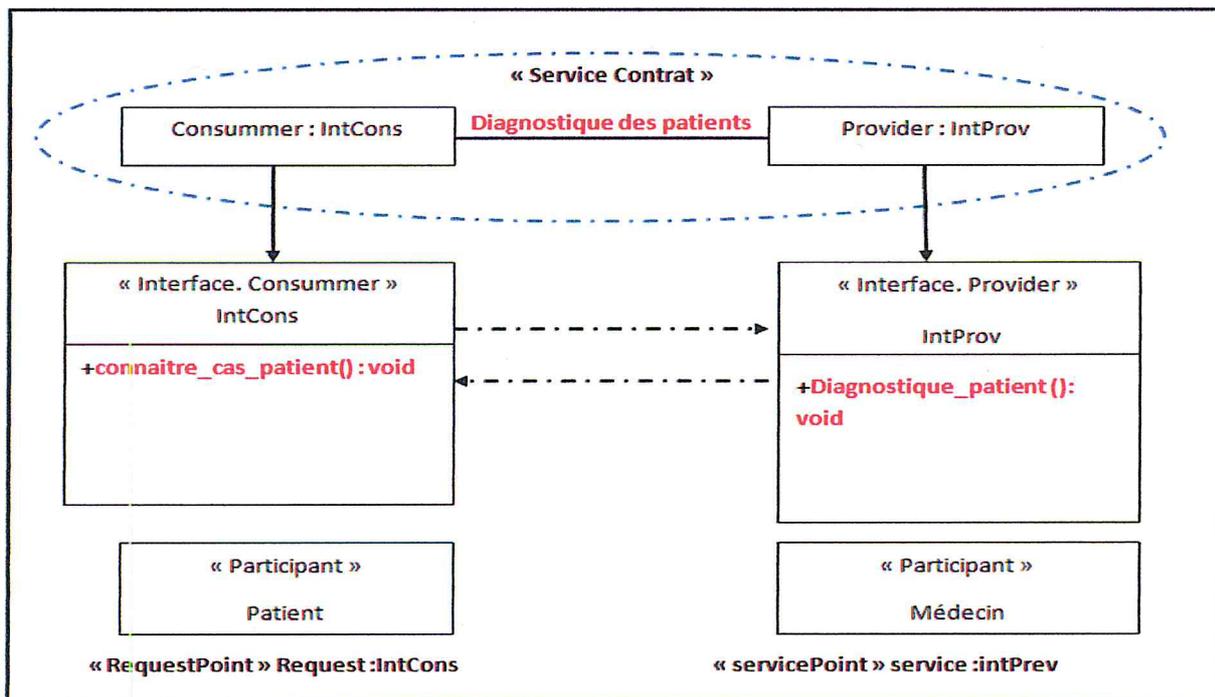


Figure I-12 Le contrat de service Diagnostic des patients.

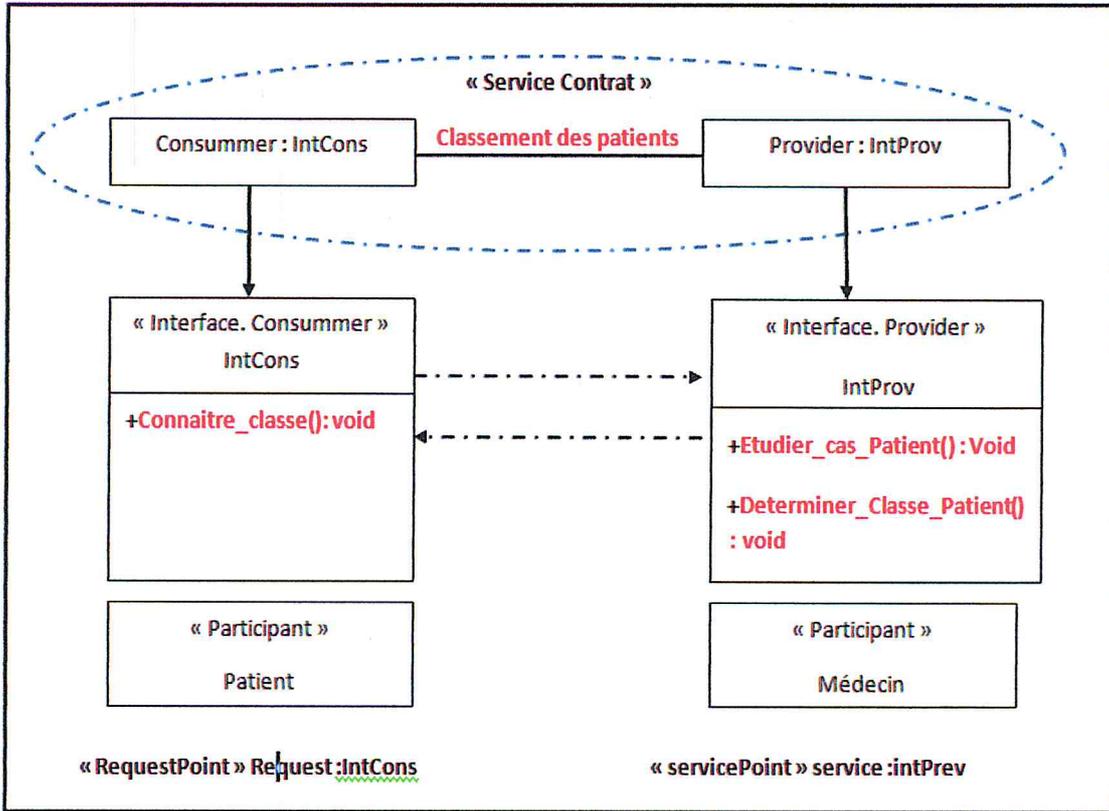


Figure I-13 Le contrat de service Classement des patients.

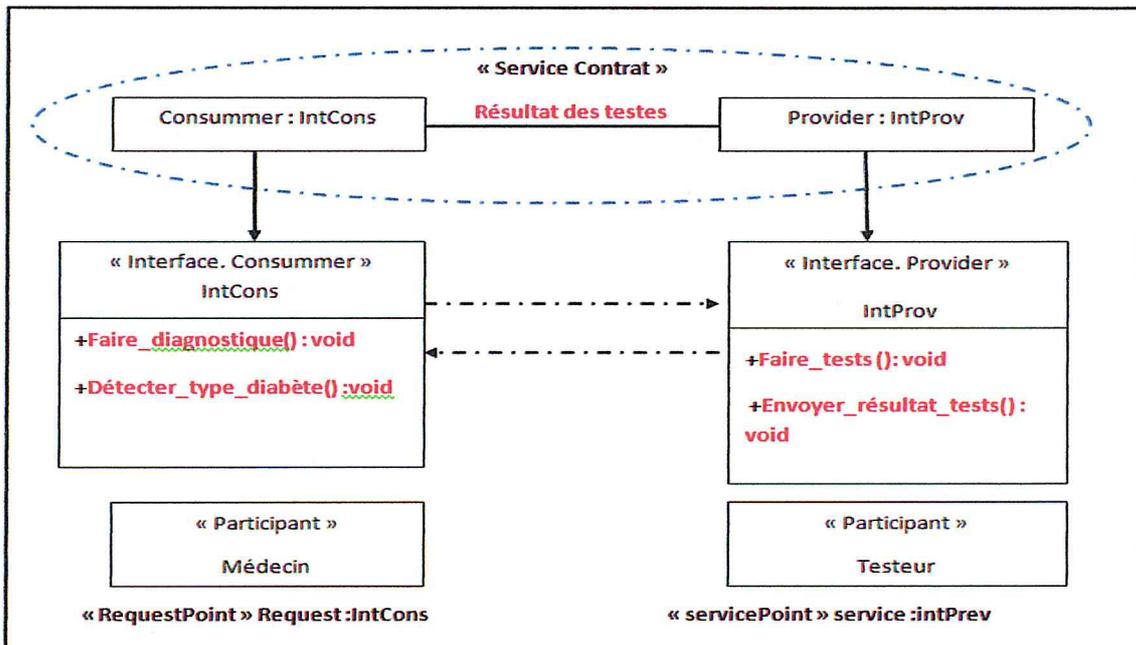


Figure I-14 Le contrat de service Résultat des testes.

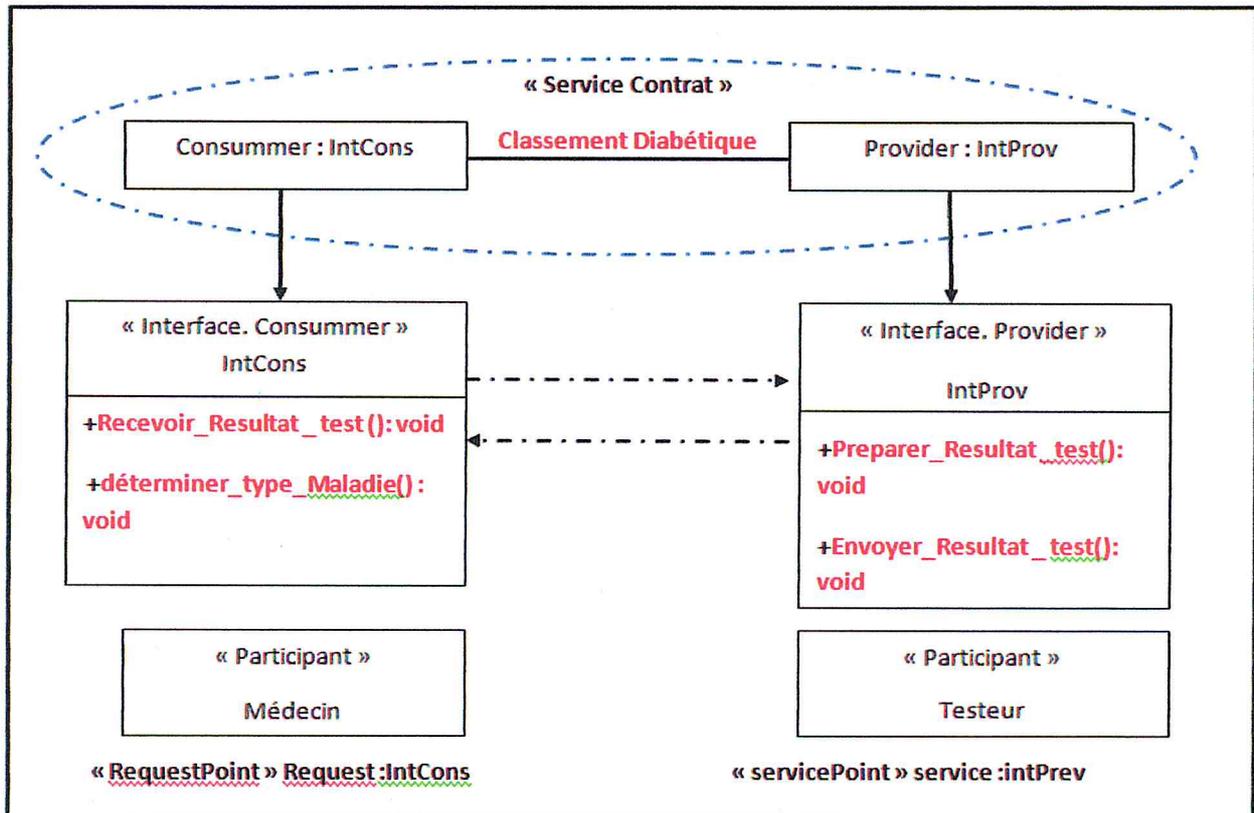


Figure I-15 Le contrat de service Classement Diabétique.

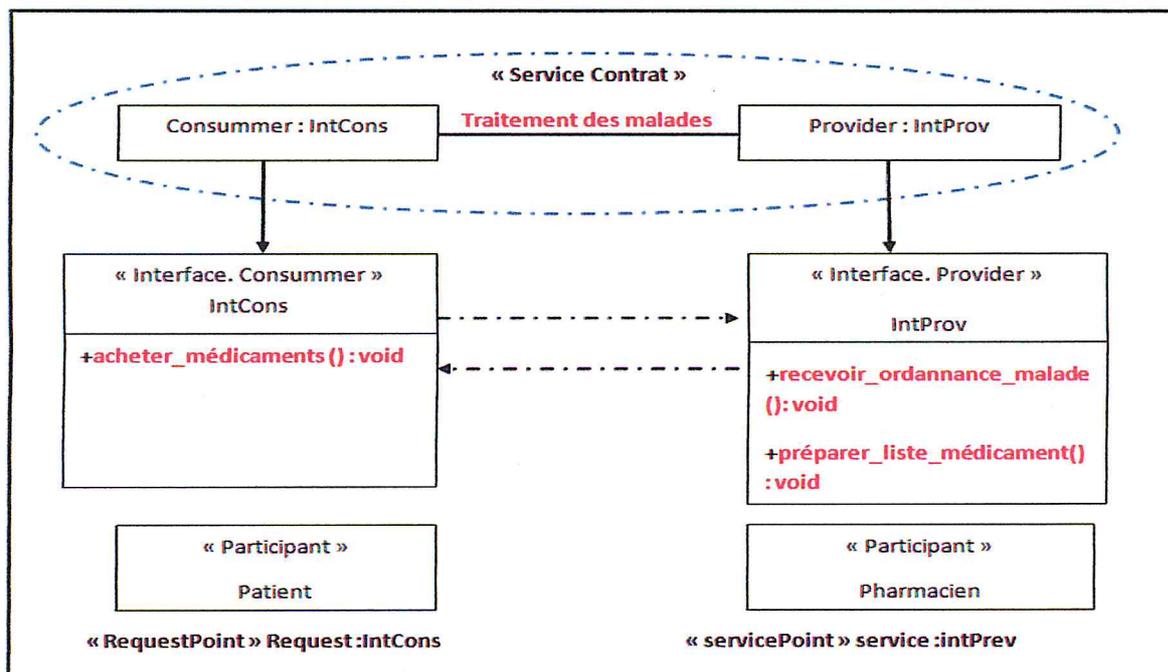


Figure I-16 Le contrat de service Traitement des Malades.

5. Phase 4: Implémentation du système

5.1 Les choix technique

Dans cette phase, nous présentons les différents choix effectués pour réaliser la Proposition.

La figure suivante présente les choix techniques qu'il faut prendre en considération.

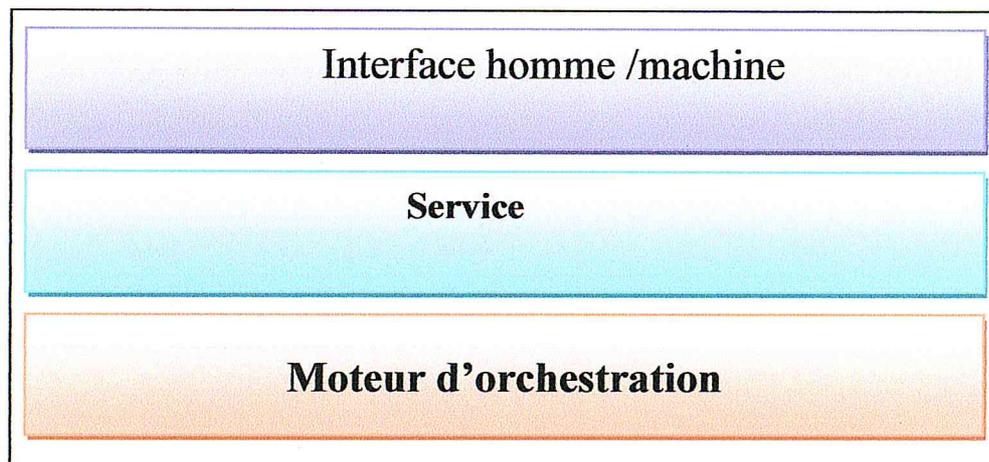


Figure I-17 Choix techniques.

5.1.1. Interface Homme/Machine

Nous avons opté à la solution **JAVA EE**.

Java Enterprise Edition, ou **Java EE** (anciennement **J2EE**), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toutes les implémentations de cette spécification contiennent un ensemble d'extensions au framework Java standard (**JSE**, *Java Standard Edition*) afin de faciliter notamment la création d'applications réparties.

5.1.2. Moteur d'orchestration

Notre choix se pointe sur **Bonita BPM**.

Bonita BPM est une suite de gestion et de workflow de processus métier open-source créé en 2001. Il a été lancé en France Institut National de Recherche en Informatique, puis il avait incubé plusieurs années à l'intérieur de la société informatique française Groupe Bull.

Depuis 2009, le développement de Bonita est pris en charge par une société dédiée à cette activité : Bonitasoft.

5.1.3. Serveur http

Le logiciel libre **Apache** HTTP Server (Apache) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Le but de ce projet est de fournir un serveur sécurisé, efficace et extensible qui fournit des services HTTP en synchronisation avec les standards HTTP actuels.

Apache a été le serveur Web le plus populaire sur Internet depuis Avril 1996.

La **Figure I-3** résume les choix techniques effectives pour réaliser le système demandé

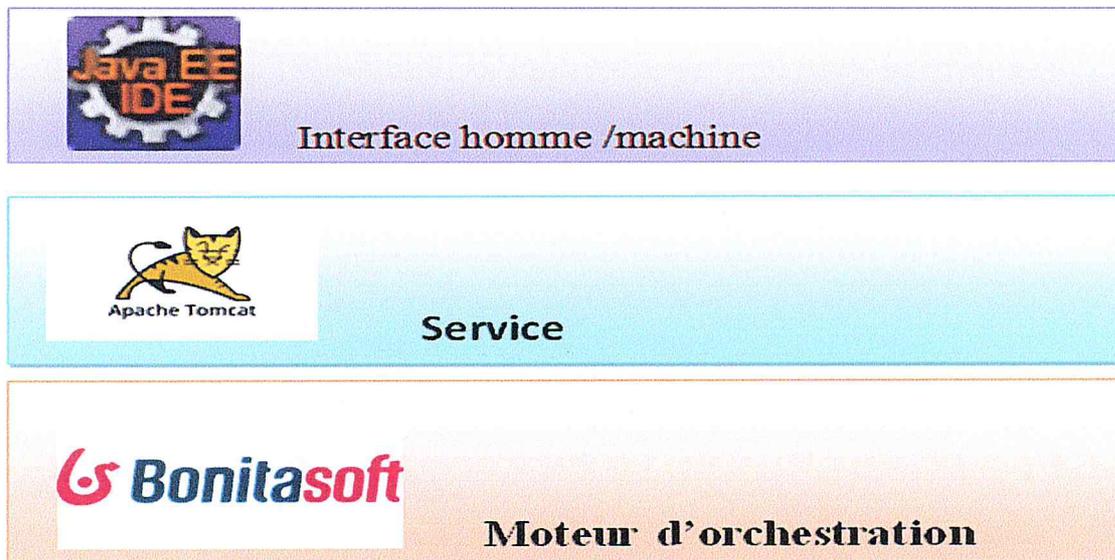


Figure I-18 une vue globale de la plateforme de destination.

5.2 L'implémentation des services

Après la modélisation nous présentons L'implémentation des services par les figures suivantes :

DIABETE INSIPIDE

Infirmier

Début

Validation des informations du patient

Que devez vous faire ?

Technique

Décisionnelle

Figure I-19 Vue arborescente des services d'infirmier.

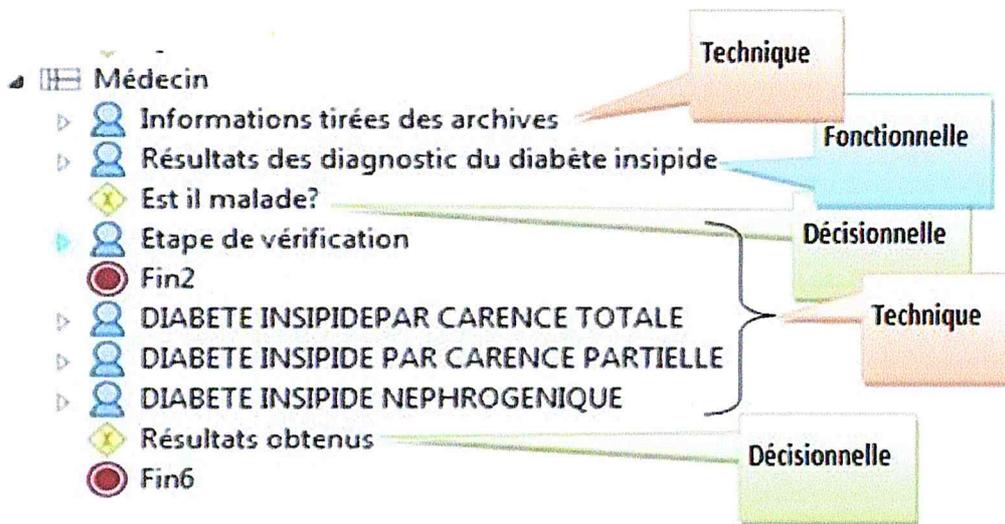


Figure I-20 Vue arborescente des services du médecin.

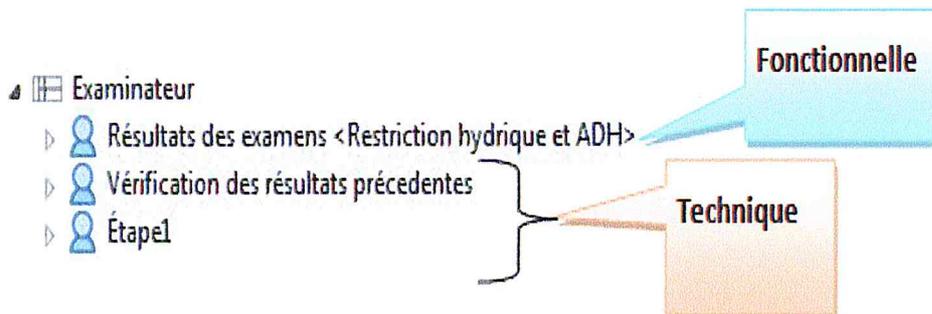


Figure I-21 Vue arborescente des services d'examineur.



Figure I-22 Vue arborescente des services du pharmacien.

5.3 Représentation du système

Afin de montrer l'exécution du système développé, nous utilisons les diagrammes de séquences pour illustrer les interactions entre les tous services.

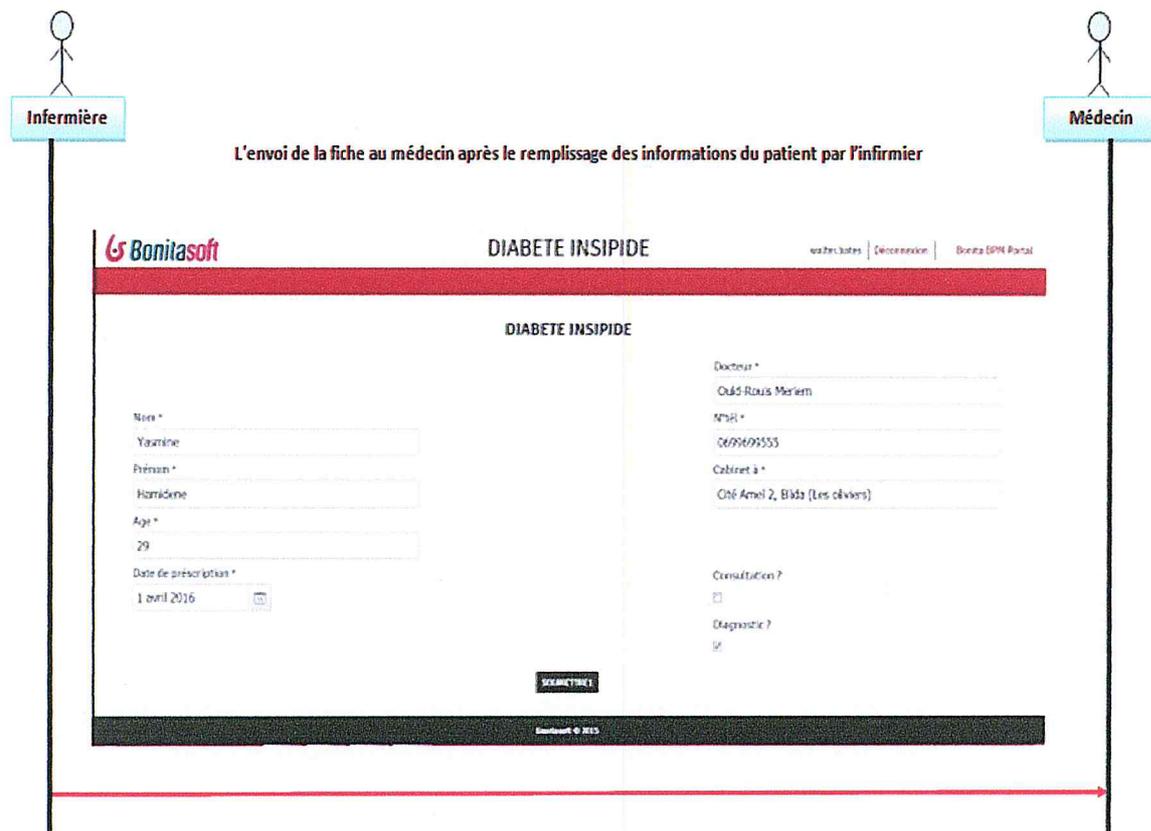


Figure I-23 Diagramme de séquence entre l’infirmier et le médecin.

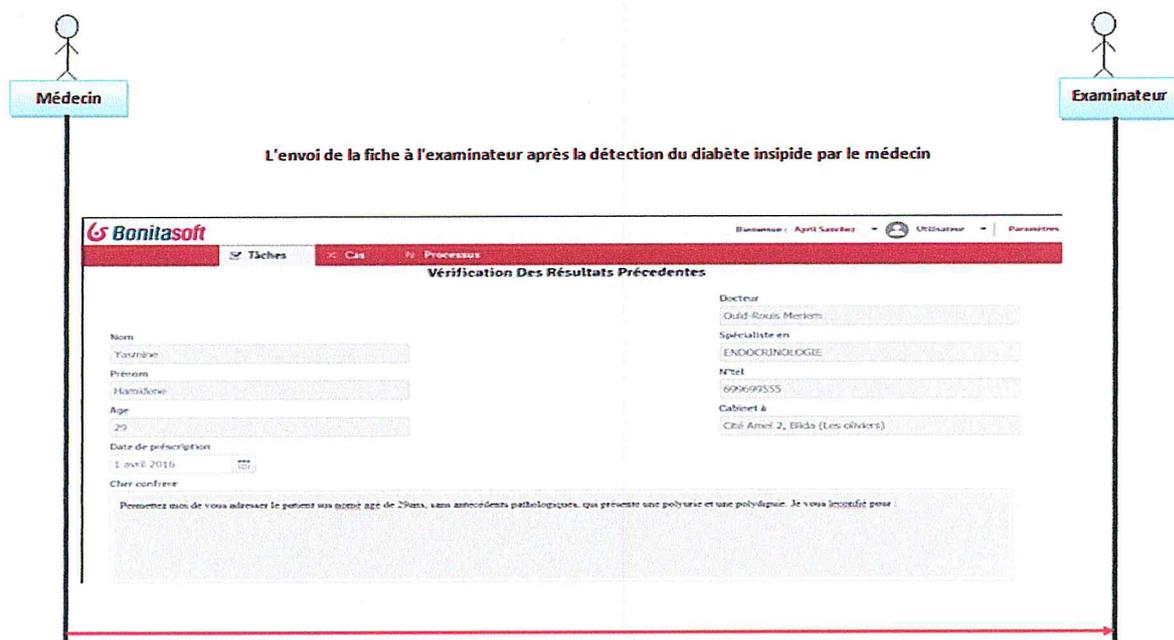


Figure I-24 Diagramme de séquence entre le médecin et l’examineur.

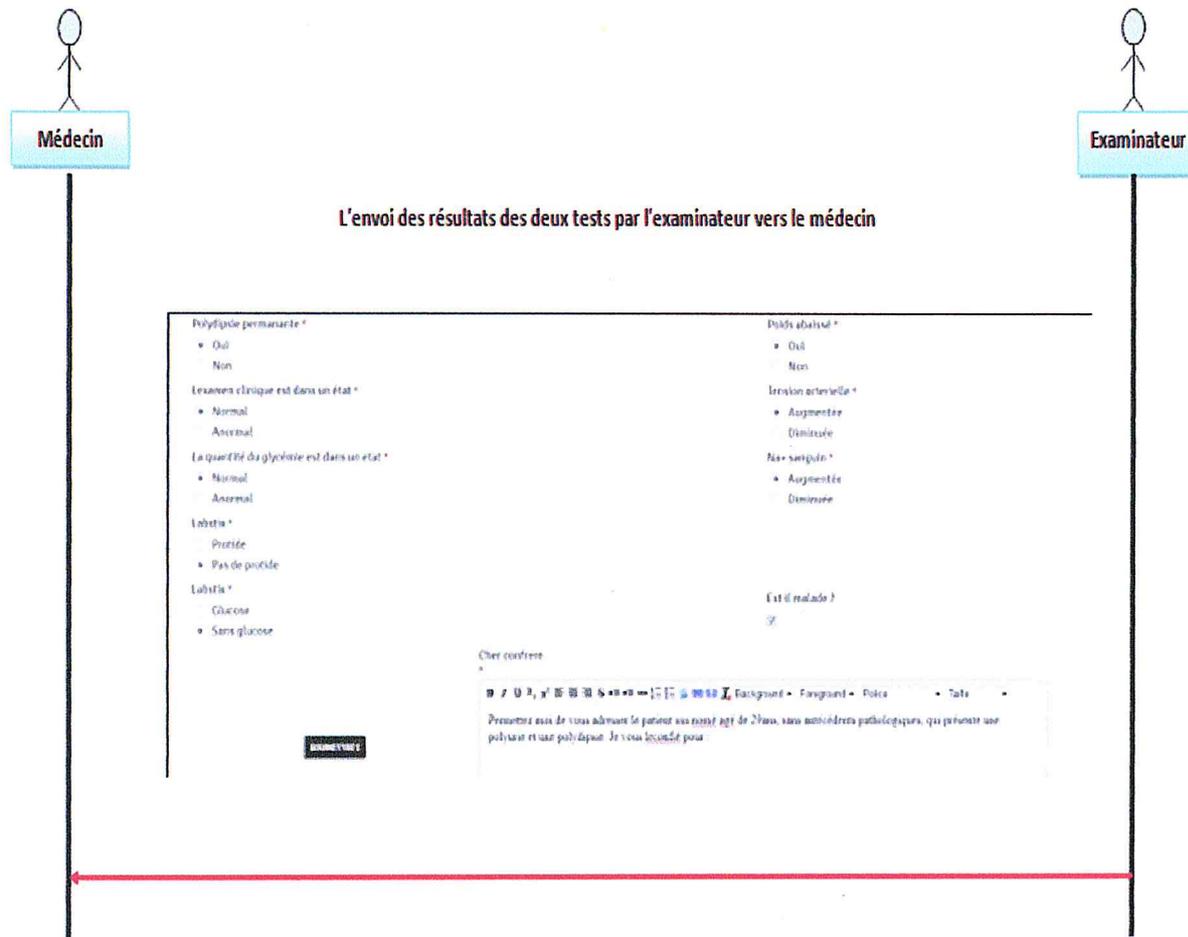


Figure I-25 Diagramme de séquence entre l'examineur et le médecin.

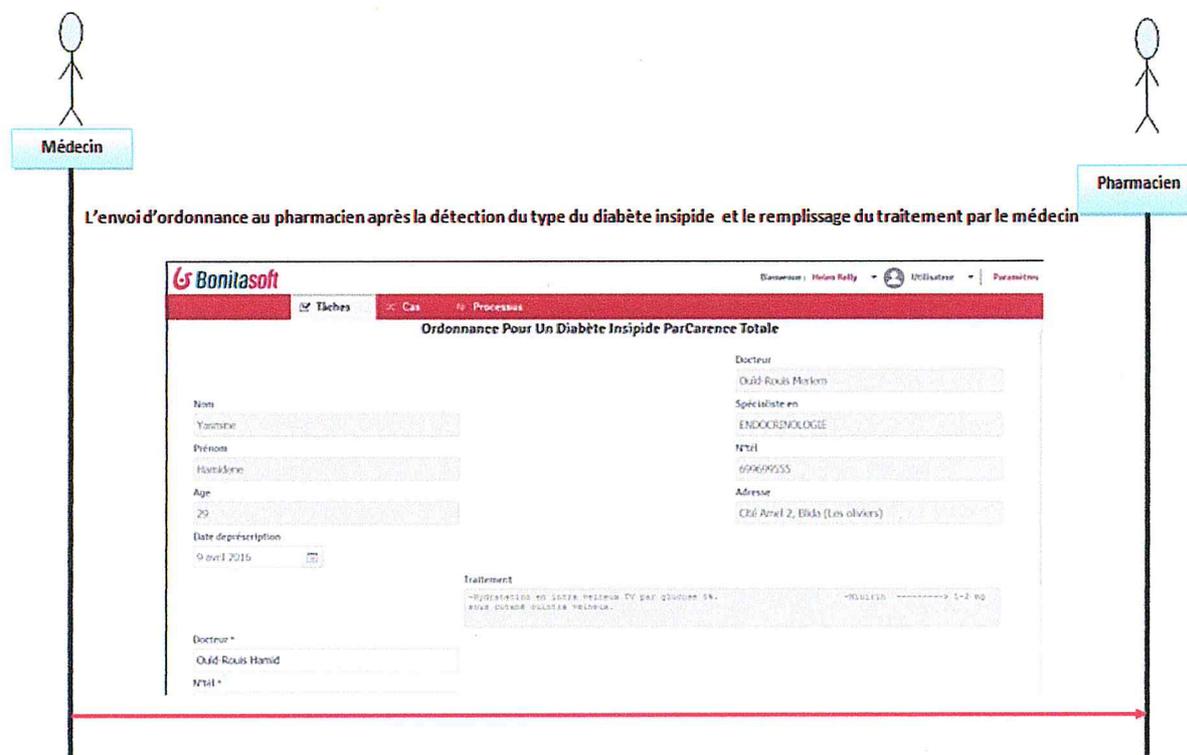


Figure I-26 Diagramme de séquence entre le médecin et le pharmacien.

6. Conclusion

Dans ce chapitre nous avons présenté la démarche de développement de notre application, en appliquant la méthode SOA+d, suivant quatre phases : phase d'analyse qui comporte les différents diagrammes (cas d'utilisation et le BPMN et le DMN), et dans la deuxième phase identification et catégorisation des services, nous appliquons les neuf règles de passage sur notre travail avec l'identification des différents services pour chaque aspect (aspect système d'information et l'aspect métier et l'aspect décisionnels), et dans la troisième phase nous présentons l'architecture globale de notre systèmes avec les différents contrats liées, et la dernière phase c'est la représentation de notre application.

Après la réalisation du système E-santé (diabète) nous avons rédigé un article scientifique (annexe), ce dernier résume le travail effectué et donne un aperçu de développement orienté service le papier sera soumis à une conférence international (ISIA 2016).

Conclusion Générale

1. Conclusion générale

Nous constatons que ce projet était une bonne occasion pour sortir du cadre théorique et appliquer les connaissances acquise lors des études universitaires de créer un programme qui annule les difficultés et le manuellement de travail.

L'objectif de ce travail était assurer l'aspect décisionnel dans le « processus d'E_santé selon une architecture orientée services ».

A cet effet, nous avons fait un travail de recherche sur le SOA (l'architecture orientée services) et l'automatisation du processus métier. Après l'étude du domaine de ce processus, nous avons conçu une méthode de classement qui classe le patient selon le type de diabète insipide.

La première partie de ce travail, c'est la partie théorique qui est composé d'une introduction générale au début et de trois chapitres, Le premier chapitre présente l'architecture orientée services ainsi, il montre quelques méthodes de mise en place d'une SOA, et dans le deuxième chapitre nous avons présenter les langages de modélisation de SOA :SOAML (Service Oriented Architecture Modeling Language), BPMN (Business Process Modeling Notation) et DMN (Decision Model and Notation), ainsi la nouvelle méthodologique SOA +d (Service Oriented Architecture with a Decision). Et en fin dans le dernier chapitre nous avons parcourir les déférentes phases de la méthodologique SOA+d pour présentons la démarche de développement de notre processus avec une conclusion générale à la fin, cette partie liée à un article (Annexe) dont récapitule tous notre processus (La partie théorique et pratique).

Pour l'utilisation d'un processus de développement adapté à notre champ d'étude nous suivons les phases de la méthode SOA+d, cette méthode vise à intégrer l'aspect décisionnel de l'architecture de type SAO au sein d'une entreprise.

Et ce qui concerne le coté pratique nous avons créé une application qu'est facile à utiliser, ainsi que diminuer la perte de l'information, perte de temps considérable lors de le recherche et les transmissions des informations des patients, et le risque d'erreur est moins élevé.

Le début de travail a été très difficile, ce qui est principalement dû à la multiplicité des concepts SOA des notations BPMN, SoaML et la nouvelle notation DMN. Mais ces difficultés nous ont permis d'acquérir un esprit de synthèse.

Finalement, nous avons pu atteindre notre objectif qui est la construction d'une application très utile et très efficace qui peut aider beaucoup de médecins et leur faciliter la tâche avec sa prise automatique de décision et son fonctionnement régulier et efficace. Elle aide également à éliminer les obstacles à distance, d'autre part nous avons pu réaliser un système qui aide même des milliers de malades éloignés géographiquement et qui souffrent des maladies chroniques. Sans oublier la valeur que nous avons rajoutée à l'informatique avec cette réalisation.

Pour réaliser ce travail nous avons utilisé Interface Homme/Machine JAVA EE, Le serveur http apache server et pour orchestrer les services nous avons adopté le moteur d'orchestration Bonita BPM, et nous avons passé un beau moment de recherche, voire plus qu'un mois pour que nous ayons trouvé ce dernier qui représente un moteur d'orchestration et un serveur d'exécution à la fois, sans oublier le temps passé pour trouver la version appropriée.

Notre application est limitée par certains services, afin qu'elle ne soit utilisée que par un médecin spécialiste en ENDOCRINOLOGIE, et pour la détection et le traitement du diabète insipide seulement.

Comme perspective nous voulons accomplir une extension de notre travail, par la réalisation d'une application avec des services utiles par des médecins en d'autres spécialités comme : pédiatrie, gynécologie, hypertension, insuffisance cardiaque ...etc. Et également pour un médecin généraliste qui demande une vaste application.

D'une autre manière, notre but ultime est la construction des services dans n'importe quel domaine afin d'informatiser et de résoudre vos problèmes, cela ne se limite pas seulement au domaine de la santé ou seulement pour un médecin spécialiste en ENDOCRINOLOGIE, notre travail sera préparé selon vos besoins.

A la fin, nous concluons notre travail et notre réalisation par la présentation d'un article (Annexe) qui résume tous notre cycle dès le début de notre mémoire jusqu'à la description et la réalisation de notre application.

annexe

Une architecture orientée service et d'aide à la décision dédiée au E-Santé.

Application : Le diabète insipide.

Ould-Rouis Meriem *, Ouslimani Djedjigua, Boumahdi Fatima

Université DE BLIDA 1 BP 270 BLIDA (09000) ALGÉRIE.

Abstract. Dans cet article, L'objectif appréhendé par le présent travail, est automatiser le processus de traitement d'un malade infecté par le diabète insipide à l'aide d'une architecture orientée services (SOA) afin de faciliter la tâche et la communication entre un médecin spécialiste en Endocrinologie, testeur et un pharmacien ; afin de nous aide même à soigner les malades éloignés géographiquement. Dans ce cadre, nous avons utilisé le diagramme de cas d'utilisation, diagramme DMN et également le diagramme BPMN, ce dernier a été utilisé pour l'exécution et la construction de nos services qui s'exécutés selon deux décisions : la première décision c'est la précision du cas du patient, s'il est infecté par le diabète insipide ou non. La deuxième décision c'est la détection du type du diabète insipide sachant qu'il existe trois types, cette décision permette au malade de prendre le traitement et au pharmacien de recevoir l'ordonnance pour terminer le long du processus.

1 Introduction

Une architecture orienté service est une architecture logicielle s'appuyant sur un ensemble de services simples. Ils sont développés en s'inspirant des processus métier de l'entreprise. Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services. Ces services représentent les fonctions basiques des fonctionnalités des entreprises. Ils dialoguent entre eux au travers de bus ou par Internet, nous parlons alors de Webservice (WSOA). Les échanges peuvent se faire de manière synchrone ou asynchrone. L'entreprise s'enrichit de services mutualisables permettant de répondre rapidement et avec souplesse aux demandes du marché. En effet, ils

correspondent à un processus métier mutualisable au niveau de l'entreprise. Cela permet les changements au niveau informatique des décisions stratégiques et tactiques de l'entreprise.

Il n'existe pas spécifications officielles pour l'architecture d'une SOA. Il peut être décrit par la notion, la description, la publication et l'invocation de services. La notion de service est une fonction encapsulée dans un composant que nous pouvons interroger à l'aide d'une requête composée d'un ou plusieurs paramètres et fournissant une ou plusieurs réponses. Idéalement chaque service doit être indépendant des autres afin de garantir sa réutilisabilité et son interopérabilité. La description de service est la manière de décrire les paramètres d'entrée du service, le format et le type des données retournées. Le principal format de description de services est WSDL (Web Services Description Language), normalisé par le W3C. Ensuite la publication consiste à ajouter dans un registre (en anglais registry ou repository) les services disponibles aux utilisateurs, tandis que la notion de découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés. Le principal standard utilisé est UDDI (Universal Description Discovery and Integration), normalisé par l'OASIS. Enfin l'invocation représente la connexion et l'interaction du client avec le service. Le principal protocole utilisé pour l'invocation de services est SOAP (Simple Object Access Protocol) (Morley et al, 2005) (Boumahdi, 2015).

Puisque notre sujet présente des services par rapport à la médecine, les services jouent un rôle important dans ce domaine.

Aujourd'hui, nous avons pu construire des services informatisés pour les médecins afin de leurs facilite la tâche et la communication entres eux et même entre un médecin et un patient à distance, et nous appelons ça la télémédecine.

La télémédecine est l'utilisation des technologies de télécommunication et d'information pour fournir des soins de santé clinique à distance. Il aide à éliminer les obstacles à distance et peut améliorer l'accès aux services médicaux qui souvent ne pas être toujours disponibles dans les

communautés rurales éloignées. Il est également utilisé pour sauver des vies dans des situations de soins et d'urgence.

Bien qu'il y avait des précurseurs lointains à la télémédecine, il est essentiellement un produit de technologies de télécommunication et d'information 20e siècle. Ces technologies permettent la communication entre le patient et le personnel médical à la fois avec la commodité et la fidélité, ainsi que la transmission des données médicales, d'imagerie et de l'informatique de la santé d'un site à un autre.

Les premières formes de télémédecine réalisée avec téléphone et radio ont été complétées par la vidéo téléphonie, les méthodes de diagnostic avancées prises en charge par des applications client / serveur distribué, et en outre avec des dispositifs de télémédecine pour soutenir les soins à domicile.

L'osque nous voulons préciser un peu plus, dans notre application et notre mémoire, nous n'avons pas présenté la télémédecine de manière générale ; nous avons choisi le processus du diabète insipide en détail.

Donc notre mémoire comporte une introduction générale suivi d'une partie théorique et une autre pratique.

La première partie sera consacrée à l'étude théorique des grands concepts rencontrés lors de la réalisation de notre projet, structurée en deux chapitres.

Le premier chapitre présente l'architecture orientée service et sa relation avec le domaine de la médecine et la santé.

Et le deuxième chapitre présente la nouvelle méthode de l'architecture orientée service c'est l'architecture orientée service décisionnelle, avec toutes ses aspects.

La deuxième partie « La conception de l'architecture orientée service » consacrés d'abord à la conception de la solution pour notre problématique avec la méthode Soa+d, tout en définissant les différents aspects de cette méthode, en utilisant les langages de modélisation SOAML (Service Oriented

Architecture Modeling Language), BPMN (Business Process Modeling Notation) et DMN (Decision Model and Notation). Ensuite, nous passerons à l'implémentation de la solution retenue au niveau de la conception après une description de l'environnement du travail. Nous achèverons cette partie avec la présentation du système réalisé.

A la fin une conclusion générale résumera les principales leçons tirées.

Nous savons bien que notre but ultime ne nécessite pas seulement la réalisation d'une architecture orienté service que nous avons présenté dans l'introduction, il faut qu'elle soit décisionnelle. C'est ce que nous allons voir par la suite :

2 SOA+D

Dans cette partie nous allons présenter la nouvelle méthodologie qui est dénommée SOA +d (Service Oriented Architecture with a Decision).

SOA +d s'adresse aux petites et moyennes entreprises pour lesquelles, il est indispensable de proposer une solution basée sur des standards. L'utilisation des standards constitue une boîte à outils à laquelle s'ajoute une démarche guidant la construction de l'architecture de services à chaque étape. En effet les standards UML (Unified Modeling Language), BPMN (Business Process Modeling Notation), DMN (Decision Model and Notation) et SoaML (Service Oriented Architecture Modeling Language) sont utilisés dans le cadre de la méthode SOA +d.

D'après ce que nous avons vu le DMN est l'un des langages de modélisations utilisés par SOA+d, le but de ce langage est la normalisation des notations pour la modélisation des décisions.

L'objectif principal de DMN est de fournir une notation commune facile à comprendre par tous les utilisateurs du métier, à partir des besoins des analystes métier pour créer les exigences de la décision initiale, puis des modèles de décision plus détaillées, pour les développeurs techniques chargés d'automatiser les décisions dans les processus, et enfin, pour les personnels de métier qui permettront de gérer et de contrôler ces décisions. DMN crée un pont standardisé pour l'écart entre la conception de la décision du métier et l'implémentation de la décision. La notation DMN est conçue pour être utilisable aux côtés de la notation des processus métier de la norme BPMN. La modélisation de décision est réalisée par des analystes du métier dans le but de comprendre et de définir les décisions utilisés dans une entreprise ou une organisation. Ces décisions sont généralement les décisions opérationnelles prises dans les processus métier au jour le jour, plutôt que la prise de décisions stratégiques pour lesquels il existe quelques règles et représentations.

Trois utilisations du DMN peuvent être distinguées dans ce contexte :

- Pour la modélisation de la prise de décision humaine.
- Pour la modélisation des exigences en matière de prise de décision automatisées.
- Pour implémenter le processus décisionnel automatisé.

3 Description détaillée de la méthode proposée SOA+d

SOA +d est une démarche méthodologique à quatre phases mettant en exergue une démarche de construction de l'architecture de services au sein de l'entreprise : la construction de la SOA métier, la construction de la SOA Informationnelle et la construction de la SOA Décisionnelle.

Chaque phase de la méthode SOA +d peut comporter une ou plusieurs étapes et peut être réalisée soit par un intervenant métier, un intervenant d'information, ou un intervenant de décision soit par les trois au même temps.

Nous allons présenter les différentes phases de SOA+d :

3.1 La Phase 1 : Analyse :

La première phase de SOA +d est réalisée conjointement par les intervenants métier, les intervenants d'information et les décideurs de l'entreprise. Elle comporte trois étapes majeures : l'élaboration du modèle métier, l'étude du système d'information et la construction d'une ou des solutions aux problèmes décisionnels.

La phase d'analyse est définie en se basant sur un ensemble de modèles permettant de décrire les exigences métiers. Principalement, cette phase se compose des modèles des cas d'utilisation, BPMN et des notations DMN. Les modèles des cas d'utilisation permettent la formalisation des besoins des utilisateurs associés aux différents acteurs interagissant avec le système. Les modèles BPMN permettent la description des différents scénarios de réalisation des cas d'utilisations. A la fin, le DMN décrit la conception des solutions des décisions.

Cette phase comprend trois étapes: l'analyse du système d'information, l'analyse métier et l'analyse de décision.

Le diagramme de cas d'utilisation pour modéliser l'aspect système d'information, le diagramme de BPMN pour modéliser l'aspect métier et le diagramme DMN pour modéliser l'aspect décisionnel.

Maintenant, nous parlons par rapport à nos problèmes qui sont le classement des malades si sont infectés par le diabète insipide ou non et le deuxième problème qui est le classement des patients infectés par le diabète insipide selon le type de ce dernier.

Nous commençons par l'aspect métier par rapport à notre sujet, nous définissons deux sous processus, sans oublier que ce classement ne se fait qu'après quelques examens :

Sous processus 1: Classement des patients.

- Classe des patients infectés par le diabète insipide.
- Classe des patients non infectés par le diabète insipide.

Sous processus 2: Classement diabétique (Diabète insipide).

- Classe du diabète insipide central par carence partiel.
- Classe du diabète insipide central par carence totale.
- Classe du diabète insipide néphrogénique.

Et maintenant nous passons à l'aspect décisionnel par rapport à notre sujet ; c'est-à-dire, il faut définir les points qui nécessitent une bonne décision pour terminer les tâches suivantes ou pour passer à un autre état, pour cela nous définissons aussi deux points ou deux cas :

Le cas des patients.

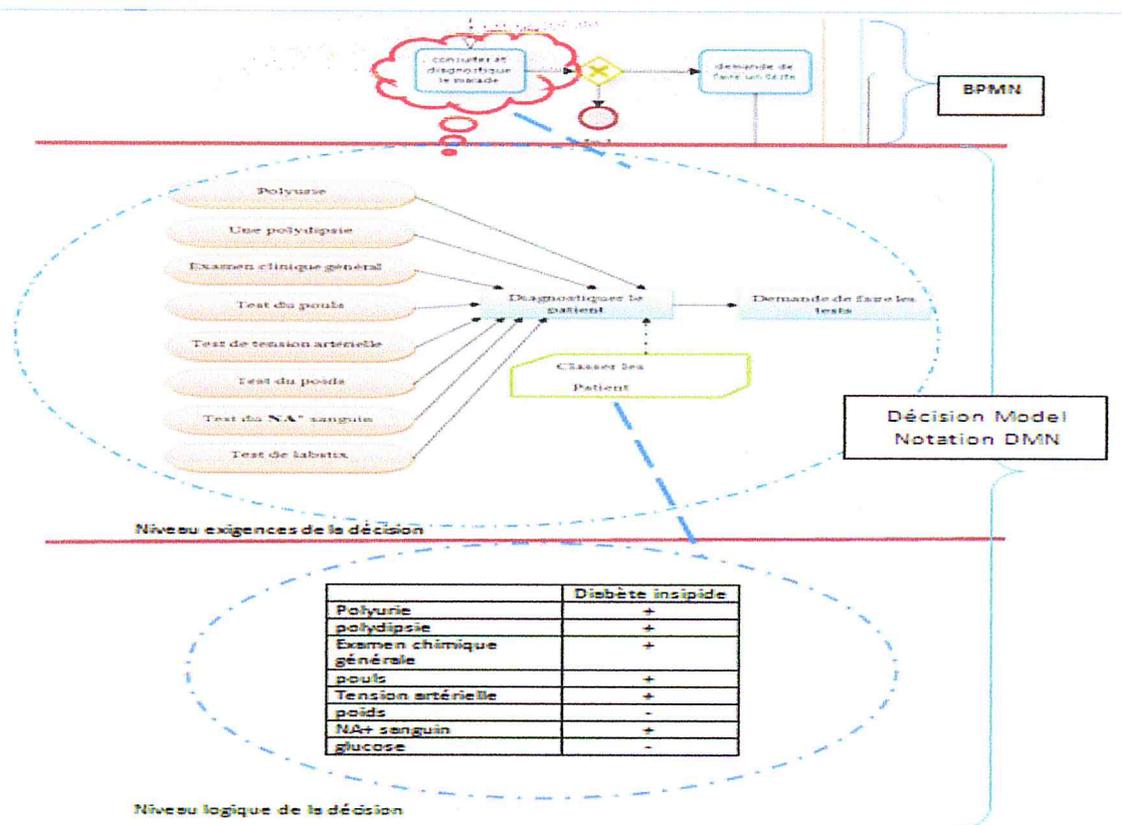
Le cas du patient comporte deux classes :

La classe des patients diabétique (Diabète insipide) : ce classement se fait après l'apparition du patient en assez bon état général dans l'examen clinique et l'apparition de certains symptômes (une polyurie et une polydipsie...etc).

La classe des patients non diabétique (Diabète insipide) : ce classement se fait après l'absence d'au moins l'un des symptômes du classement précédent.

La Figure suivante montre le DMN de la décision « classement des patients ».

Fig.1.classement des patients DMN.



	Diabète insipide
Polyurie	+
Polydipsie	+
Examen clinique générale	+

Pouls	+
Tension artérielle	+
Poids	-
NA+ sanguin	+
Glucose	-

Voici l'explication logique appropriée à ce diagramme :

Explication logique

POLYURIEpos : augmentation du volume des urines.

POLYDIPSIEpos : le patient boit sans cesse.

EXAMnor : l'Examen clinique est normal.

POULpos : le pouls est augmenté.

TENTIneg : tension artérielle diminuée.

POIDneg : poids abaissé.

NApos : NA+ sanguin est augmentée.

GLYCEMIEnor : la quantité de la glycémie est dans état normal.

GLUCOSEneg : pas de glucose dans labstisc.

X : patient.

La figure suivante illustre la prise de décision « Le cas des patients » par les formules logiques :

$x/POLYURIEpos(x)^{POLYDIPSIEpos(x)^{EXAMnor(x)^{POULpos(x)^{TENTINneg(x)^{POIDneg(x)^{NApos(x)^{GLYCEMIEnor(x)^{GLUCOSEneg(x)} \Rightarrow Diabetes\ insipid(x)}$

La précision du type du diabète insipide.

La précision du type du diabète insipide comporte trois classes (trois types de diabètes insipides) :

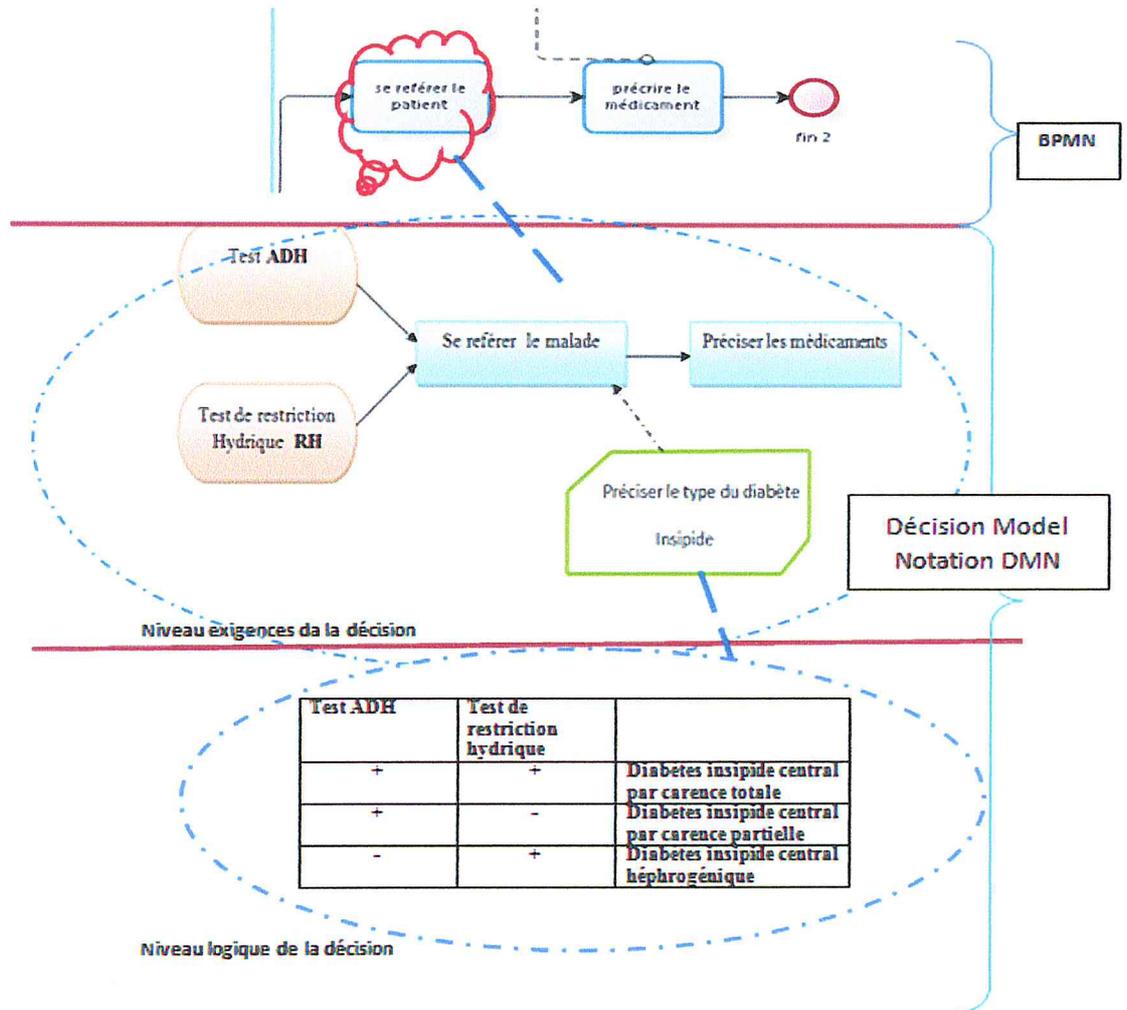
Diabète insipide central par carence totale : l'appartenance à cette classe se fait après un résultat positif (+) du test ADH et un résultat positif (+) du test de la restriction hydrique.

Diabète insipide central par carence partielle : l'appartenance à cette classe se fait après un résultat positif (+) du test ADH et un résultat négatif (-) du test de la restriction hydrique.

Diabète insipide central par carence totale : l'appartenance à cette classe se fait après un résultat négatif (-) du test ADH et un résultat négatif (-) du test de la restriction hydrique.

La Figure suivante montre le DMN de la décision « Le classement des malades selon leurs types du diabète insipide ».

Fig. 2. Le classement des malades selon leurs types du diabète insipide DMN.



Test ADH	Test de restriction hydrique	
+	+	Diabètes insipide central par carence totale

+	-	Diabète insipide central par carence partielle
-	+	Diabète insipide central néphrogénique

Explication logique

ADHpos : le résultat du test d'ADH est positif.

RHpos : le résultat du test de la restriction hydrique est positif.

ADHneg : le résultat du test d'ADH est négatif.

RHneg : le résultat du test de la restriction hydrique est négatif.

X : patient.

La figure suivante illustre la prise de décision « Classement diabétique » par les formules logiques du premier ordre :

$x/ADHpos(x)^{RHpos(x)} \Rightarrow Diabète\ insipide\ central\ par\ carence\ totale(x)$

$x/ADHpos(x)^{RHneg(x)} \Rightarrow Diabète\ insipide\ central\ par\ carence\ partielle(x)$

$x/ADHneg(x)^{RHpos(x)} \Rightarrow Diabète\ insipide\ central\ néphrogénique(x)$

3.2 La Phase 2 : Identifier et catégoriser les services

Cette activité vise à identifier les services nécessaires à l'exécution du processus métier en cours de développement et c'est la première clé dans notre méthode. Les principales entrées de cette phase sont le BPMN, les cas d'utilisation et le DMN.

La découverte des services vise à apporter un soutien dans l'identification des services qui seront utilisés par les processus métier. Afin d'identifier et de décrire les services à partir des informations fournies par les cas d'utilisation, BPMN et le DMN. Nous proposons des règles de passage comme un guide pour identifier les services (Boumahdi, 2015) .

3.3 La Phase 3 : Modélisation de Services

Elle se concentre sur la spécification des services (modèle de données, interfaces, etc.). Dans un projet SOA, il est possible de se retrouver confronté à la problématique suivante : comment modéliser les services et leurs interactions ? La notation SoaML offre une solution standardisée et basée sur l'UML et centrée sur la notion d'architecture.

Dans cette phase, les services doivent être modélisés avec un formalisme, nous adoptons une spécification à la base du langage SoaML qui offre un haut niveau d'abstraction, il est alors nécessaire d'affiner les services pour les rendre propres à une plate-forme donnée. SoaML permet d'axer la modélisation sur :

- L'identification des services, leurs exigences et leurs dépendances.
- La spécification d'un service : capacité, protocoles, politiques et informations transitant entre un service client et un service fournisseur.

À ce niveau le problème consiste à identifier les services de la solution SOA. Le but est d'identifier les services métiers publics à partir des exigences exprimées avec les processus. La solution la plus simple consiste à créer un seul service et d'y mettre toutes les opérations découvertes. Dans ce cas tous les participants deviennent dépendent

de ce même service. Cette solution entraîne un couplage fort entre les participants. Tout changement par le fournisseur du service affectera l'ensemble des participants qui le consomment.

SoaML.

En proposant SoaML (Service Oriented Architecture Modeling Language) l'OMG a adapté UML qui est fortement imprégné des principes de l'orienté objet aux principes de l'orienté service (LEMRAËT Y. , 2012).

La spécification SoaML définit un langage de modélisation des services au sens SOA.

SoaML est avant tout destiné à modéliser des services dans des environnements distribués en prenant en compte les deux vues : métiers et IT. Le profil SoML permet de personnaliser les abstractions UML nécessaires à la modélisation des services. Il clarifie comment utiliser les métaclasse UML pour modéliser une solution SOA afin de garantir que les modèles des services reflètent les cinq principes SOA suivants : couplage faible, abstraction, réutilisation, autonomie et composition (LEMRAËT Y. , 2012).

La démarche de modélisation avec SoaML.

D'abord, il faut commencer par identifier les participants à partir des processus BPMN.

Ensuite, il faut distinguer les capacités associées à chaque processus et à l'ensemble de la collaboration a fin de déterminer les rôles et les responsabilités de chaque participant.

Puis, il faut spécifier les types d'interactions de la collaboration (i.e., unidirectionnelle ou bidirectionnelle).

Ainsi, l'association des capacités aux interactions permet de déterminer les Consumers et les Providers. Enfin, il convient de concevoir les contrats de

service qui sont nécessaires à la spécification de l'architecture globale des services publics.

Architectures de services.

Est une description de la manière dont les participants (acteurs) s'échangent des services exprimés par les contrats de services.

La figure suivante présente l'architecture de service correspondante à notre travail :

Fig. 3. Architecture générale.

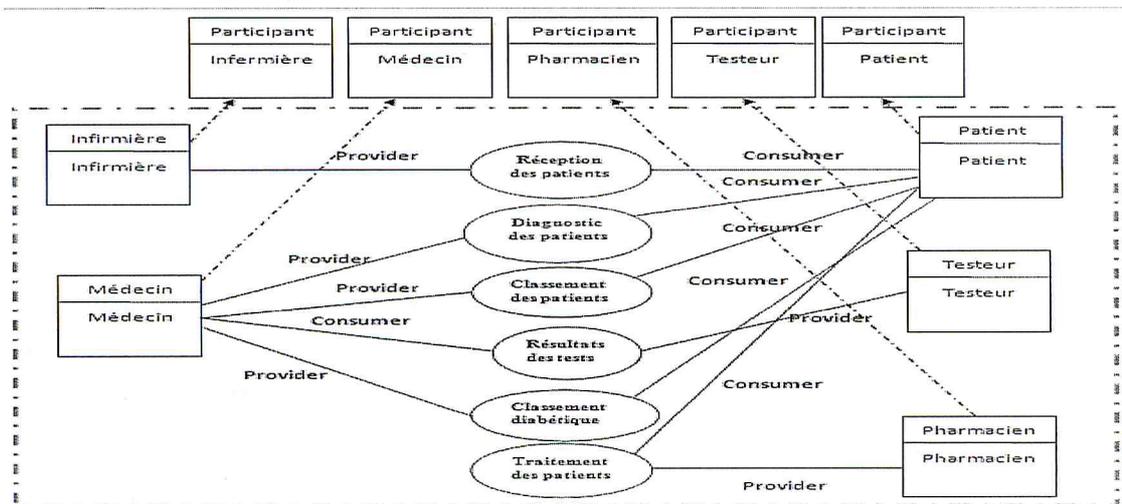


Fig. 4. Le contrat de service Diagnostic des patients.

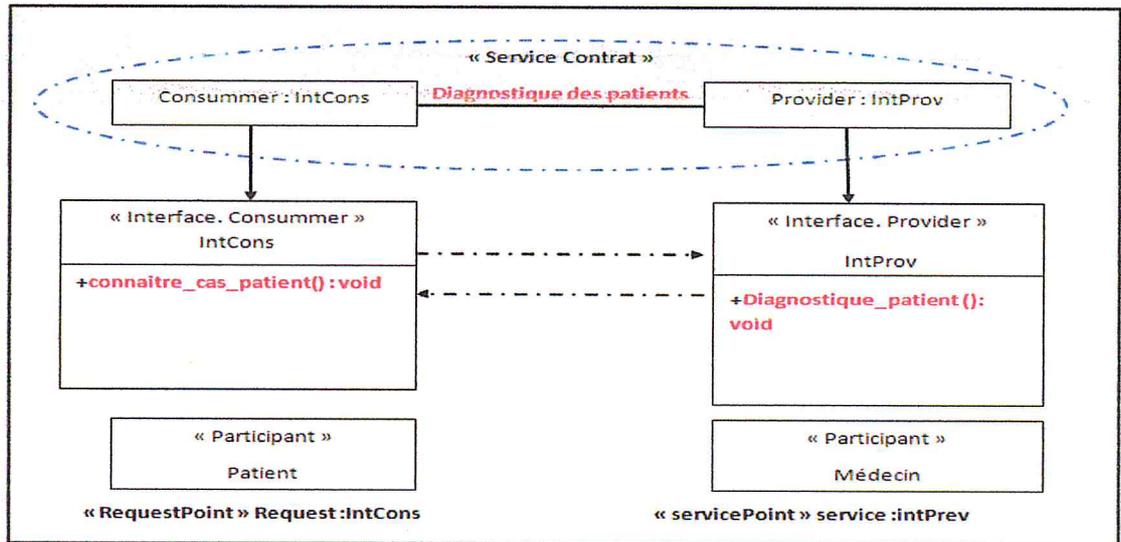
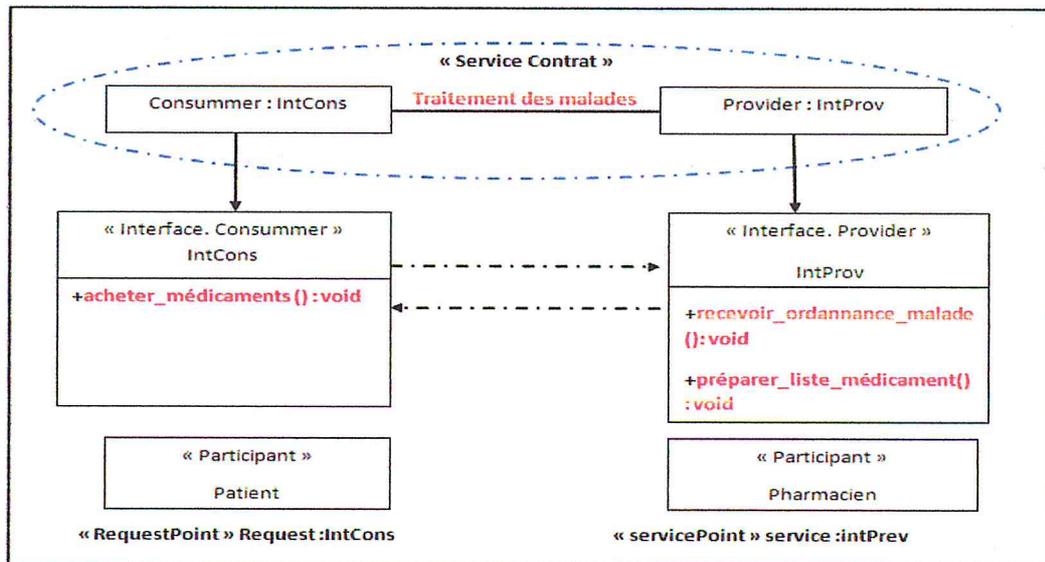


Fig.5 Le contrat de service Traitement des Malades.



Alors nous passons à la dernière phase, dans cette phase nous présentons notre travail pratique, d'une autre façon nous présentons notre système ou le logiciel que nous avons réalisé pour avoir un ensemble de services web, définissent un ensemble de tâche entre un médecin, un infirmier, un examinateur et un pharmacien. Chaque membre est responsable d'une ou de plusieurs tâches de ce système.

3.4 Phase 4: Implémentation du système

Cette phase donne un aperçu des composants SOA et les options disponibles pour la mise en œuvre d'une architecture SOA à travers les Systèmes Open Source (SOS).

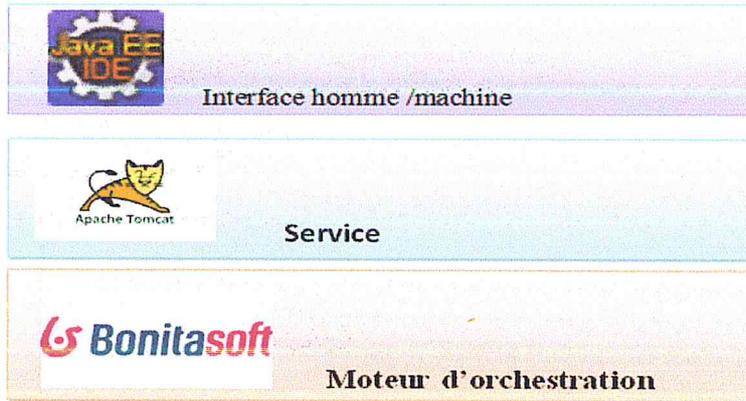
Nous résumons les différents choix technique qui doivent être pris en considération pour implémenter une architecture SOA :

- Couche Accès Service.
- Couche Service.
- Couche Service Application.
- Couche Intégration.
- Couche Application.
- Système d'exploitation.
- Service Exécution.

Les choix technique.

Dans cette phase, nous présentons les différents choix effectués pour réaliser la Proposition.

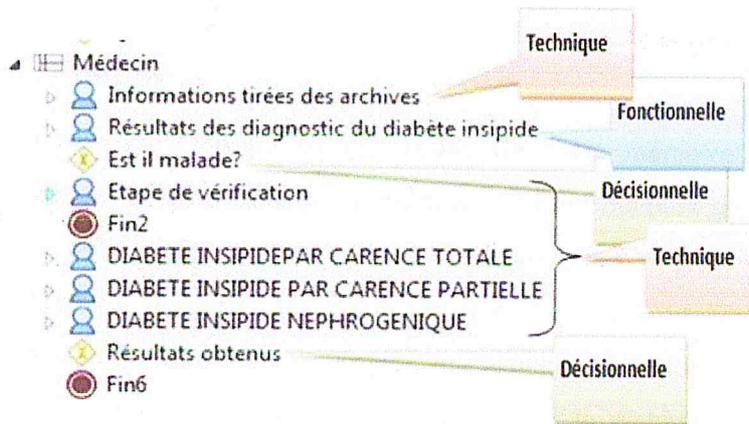
Fig. 6. une vue globale de la plateforme de destination.



L'implémentation des services.

Voici un exemple de notre implémentation :

Fig. 7. Vue arborescente des services du médecin.



Représentation du système.

Voici l'une des représentations de notre système :

[1]Y .LEMRAHET, Ecrivain, *Proposition d'une méthode de spécification d'une architecture orientée service dirigée par métier dans le cadre d'une collaboration inter-organisationnelle*. [Performance].2012.

[2]F.BOUMAHDI, *Une méthode pour intégrer l'aspect décisionnel dans L'architecture orientée service*.2015.

La bibliographie

Bibliographie

- Y .LEMRAHET, Ecrivain, *Proposition d'une méthode de spécification d'une architecture orientée service dirigée par métier dans le cadre d'une collaboration inter-organisationnelle*. [Performance].2012.
- F.BOUMAHDI, *Une méthode pour intégrer l'aspect décisionnel dans L'architecture orientée service*.2015.
- Brain Elvesaeter .D.P.(2010).*Alignig Business and IT models in service-oriented architectures using BPMN and SOAML*.
- F .Boumahdi , R. Chalal ·A.Guendouz ,K. Gasmia .(2015)*a new way to design the decision in SOA—based on the new standard Decision Model and Notation (DMN)*.
- Arsanjani A, Ghosh S, Allam A, Abdollah T, Ganapathy S, Holley K .(2008) .*Soma: a method for developing service-oriented solutions*.*IBM Syst J* 47(3):377–396.
- OMG(2012) *Service oriented architecture modeling version 1.0.1*.*Technical report, OMG*.
<http://www.omg.org/spec/SoaML/1.0.1/PDF>
- Elvesæter B, Berre AJ, Sadovykh A (2011) *Specifying services using the service oriented architecture modeling language (soaml)-a baseline for specification of cloud-based services*. In: *CLOSER*,pp 276–285.
- De Castro V, Marcos E, Vara JM (2011) *Applying cim-to-pim model transformations for the service-oriented development of information systems*. *Inf Softw Technol* 53(1):87–105.
- Erl T (2005) *Service-oriented architecture: concepts, technology, and design*. Prentice Hall PTR, Upper Saddle River.
- Herrera VV, Bepperling A, Lobov A, Smit H, Colombo A, Lastra J (2008) *Integration of multi-agent systems and service-oriented architecture for industrial automation*. In: *Industrial informatics,2008. Indin 2008. 6th IEEE international conference on. IEEE*, pp768–773